Practical No. 1 and 2: Setup a Java programming development environment and test using small program.

I. Practical Significance:

Java is the popular platform, which is used to develop various applications for the systems as well as embedded devices like mobile, laptops, tablets and many more. It is an object-oriented programming language Students will be able to setup Java environment for executing Java programs, using command prompt or using different IDEs like Eclipse, JCreator and test the setup using small java program.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and Teamwork:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java"

The practical is expected to develop the following skills:

1. Set up Java Environment for executing Java programs.
2. Execute simple program by setting path variable .

Setup a java programming development environment.

1. Using Command prompt
2. Using IDEs.

IV. Relevant Course Outcome(s)

Develop programs using Object Oriented methodology in Java.

V. Practical Outcome (PrOs)

Setup a java programming development environment.

VI. Relevant Affective domain related Outcome(s)

1. Follow safety practices.
2. Practice good housekeeping
3. Demonstrate working as a leader/ a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

Java language is compiled and interpreted.

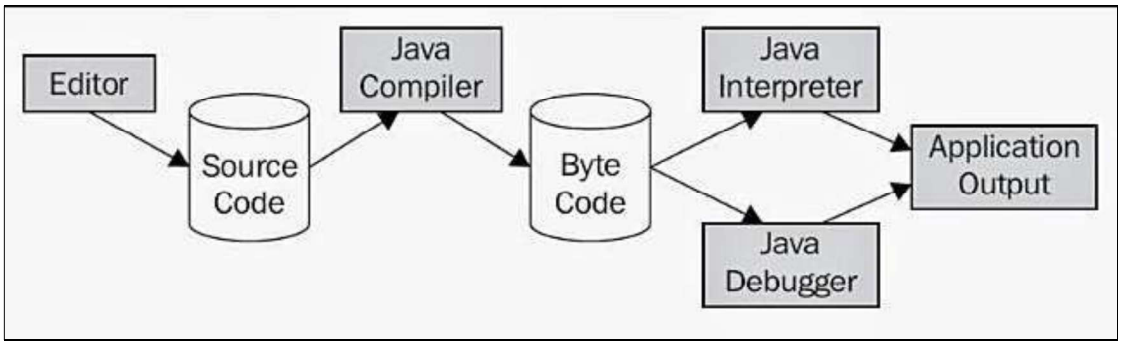


Fig. 1 Java Program Execution

1. **Procedure**

Installation for Java Software:
(On a PC loaded with windows OS – 2000/2003/2007 onwards with notepad)

1. Download JDK (jdk 1.4.0 onwards)
   Visit the
   https://www.oracle.com/technetwork/java/javase/downloads/index.html
   Download the windows version to suitable folder.
2. Double click the setup file.
3. Follow onscreen instruction.
4. When the setup is done, the complete screen appears, click on the 'Finish' Button. This completes the installation of JDK. To ensure the JDK installation / to determine the java version, type the following command at the MS Dos prompt:
   <system prompt> java –version
   It should show the output similar to following

Fig. 2 Java Version

If not then set 'path' environment variable
1. Go to start -> control panel -> system
2. System properties dialogbox will appear.
3. Select 'advanced tab' -> environment variables.
4. In the system variable list, select path and click 'edit'
5. Edit the system variable dialogbox appears. In the variable value field , append the path to the JDK bin directory (generally 'c:\program files\java\jdk1.6.0\bin' at the end. Use semicolon to separate the path of bin directory from the rest of values already available. Click 'Ok'.
6. Similarly set 'classpath' environment variable

Fig. 3 Environment variable

Note: Follow the similar instructions for other platforms (say Unix, Linux,Mac) with appropriate jdk download.

2. **Using an Eclipse for Java**
   Eclipse (@ www.eclipse.org) is an *open-source* Integrated Development Environment (IDE) supported by IBM. Eclipse is popular for Java application development (Java SE and Java EE) and Android apps.
   Installing Eclipse 4.7.2 (Oxygen 2) for Java Developers
   To use Eclipse for Java programming, you need to first install Java Development Kit (JDK).
   1. Download Eclipse from https://www.eclipse.org/downloads. Under "Get Eclipse Oxygen" ⇒ Click "Download Packages".
   2. To install Eclipse, unzip the downloaded file into a directory (e.g., "d:\myproject").

3. **Testing setup using small program:**
   Steps for editing and executing java program:
   Using an editor (e.g. Notepad)
   1. Open notepad
   2.Write the program (called java source code) in notepad
   3.Save the file as 'filename.java' in some directory. The filename must be same as the classname containing main() method.
   4.Open MS-Dos prompt.
   5. Change the directory containing to the one containing the program.
   6.Compile the program by using the command **javac<filename.java>**

7.Execute/ Run the program by using the command **java <filename>**

4. Using Eclipse

1. Launch Eclipse by running "eclipse.exe" from the Eclipse installed directory.
2. Choose an appropriate directory for your *workspace*
3. To create a new Java project using "File" menu ⇒ "New" ⇒ "Java project"
4. In "JRE", select "Use default JRE (currently 'JDK9.0.x')". But make sure that your JDK is 1.8 and above.
5. In "Project Layout" menu, select "Use project folder as root for sources and class files".
6. Push "Finish" button.
7. In the "Package Explorer" (left pane) ⇒ Right-click on "FirstProject⇒ New ⇒ Class.
8. Write a program
9. Compile and execute program.
10. Observe output on the console panel.

Sample program:

```
Class HelloWorld
{
public static void main(String args[] )
  {
System.out.println("Welcome to Hello World program");
  }
}
```


5. Output:

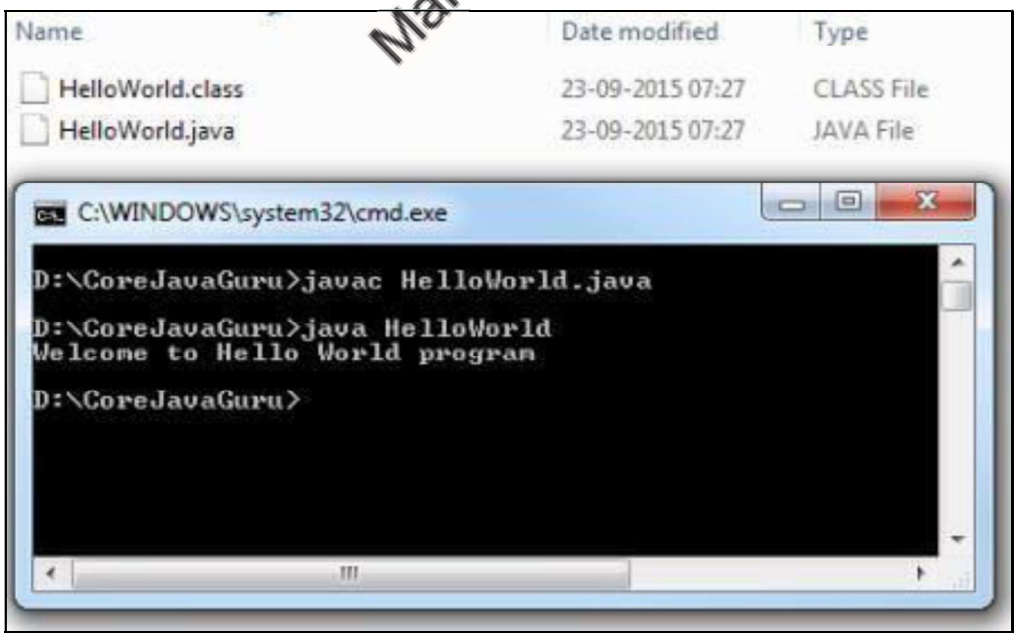


Fig 4. Output of the Program

VIII. Resources required

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity	Remark
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	As per batch size	For all Experiments
2	Operating system	Windows / Linux		
3	Software	jdk1.8.0 or above.		

IX. Resources used

S. No.	Name of Resource	Broad Specification	Qty	Remarks (If any)
1	Computer System with broad specifications	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	As per batch size	
2	Software	jdk1.8.0 or above.		
3	Any other resource used			

X. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Write installation directory path of your directory?
2. Write value of path environment variable?
3. List folders created after installation.
4. Main method is declared as static. Justify.
5. Program is named with class containing main method. Justify.

(Space for answer)

1) c:\program file (x86)\Java\ JDK 1.8.0-60\bin

2) c:\program file (x86)\Java\ JDK 1.8.0-60\bin

3) 1) bin 2) conf 3) include 4) jmods 5) legal 6) lib

4) Java main() method is always static, so that compiler can call it without the creation of an object or before the creation of an object of the class. In any Java program, the main() method is the starting point from where compiler starts program execution. So, the compiler needs to call the main() method

5) All Java programs must have an entry point, which is always the main() method. Whenever the program is called, it automatically executes the main() method first.

The main method appears in every program inside the class that is part of application But, if application is completed containing multiple files. It is comman method which get execute called as generally Main( ) method.

Practical No. 3: Develop programs to demonstrate use of if statements and its different forms.

I. Practical Significance:

In computer science, conditional statements, expressions and constructs are perform different computations or actions depending on whether boolean condition evaluates to true or false. Students will be able to use various forms of if statements to check the condition.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java".

The practical is expected to develop the following skills:

1. Write a program to use simple if statements to check conditions
2. Develop a program to use different forms of if to check multiple conditions.

IV. Relevant Course Outcome(s)

Develop programs using Object Oriented methodology in Java.

V. Practical Outcome (PrOs)

Develop programs to demonstrate use of if statements and its different forms.

VI. Relevant Affective domain related Outcome(s)

1. Follow safety practices.
2. Practice good housekeeping
3. Demonstrate working as a leader/ a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

Decision making in Java programming

Control statements are used to control the flow of execution of program based on certain conditions. These are used to cause the flow of execution to advance and branch based on changes to the state of program.

Java Selection Statements:

1. if
2. if-else
3. nested-if
4. if-else-if ladder

1. **if:** if statement is simple decision-making statement. It is used to decide whether a certain statement or block of statements will be executed or not. i.e. if a certain condition is true then the block will be executed otherwise not.
   Syntax:
   ```
   if(condition)
   {
    // Statement to execute if the condition is true;
   }
   ```
2. **if-else:** The if statement alone tells us that if a condition is true it will execute a block of statements and if the condition false, else block will be executed.
   Syntax:
   ```
   if(condition)
   {
    // Statement to execute if the condition is true;
   }
   else
   {
    // Statement to execute if the condition is false;
   }
   ```
3. **nested-if:** A nested if is an if statement that is the target of another if or else. Nested if statements means an if statement inside an if statement.
   ```
   if(condition1)
   {
      // execute when condition1 is true.
   if(condition2)
       {
          // execute when condition2 is true.
       }
   }
   ```
4. **if-else-if ladder**: A user can decide among multiple options. The if statements are executed from top down. When one of condition is true, the statement associated with that if is executed, and the rest of the ladder is bypassed. If none of the conditions is true, then the final else statement will be executed.

VIII. Resources required (Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1				
2				

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Write any program to check multiple conditions using if statement.

```
public class IfStatement
 {
public static void main(String[] args)
 {
   int number = 10;
   if (number > 0)
 {
     System.out.println("The number is positive.");
   }
     System.out.println("Statement outside if block");
  }
}
```


XI. Result (Output of Code):

```
The number is positive.
Statement outside if block
```

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. List operators used in if conditional statement.
2. In if-else construct which part will be executed if condition is true.
3. State the condition when the else part will be executed with example.

4. Which of the following operator is used in if:
   a. Assignment operator (=) b. comparison operator (= =)

(Space for answer)

1)

< ~Less than

> ~Greater than

<= ~Less than or equal to

>= ~Greater than or equal to

== ~Equality operator

2) If condition is true then off course if conditional part of executed .

3) Example :

```
class Main {
public static void main(String[]
 int number = -5;
if (number > 0) {
System.out.println("The number is positive.");
}
else {
System.out.println("The number is not positive.");
}
System.out.println("Statement outside if...else block");
   }
}
```


Here, the value of number is -5. So the test expression evaluates to false.
Hence, code inside the body of else is executed.

4) b. comparison operator (==) is used in if condition .

XIII. Exercise:

1. Write output of code in the given space.

Sr. No.	Program Code	Output

1.	public class NestedIfExample {
	
	public static void main(String args[]){
	int num=70;
	if(num< 100){
	System.out.println("number is less than 100");
	if(num> 50){
	System.out.println("number is greater than 50");
	}
	}
	}
	}
		Number is less than 100
		
		Number is greater than 50

2.	class IfStatement {
	public static void main(String[] args) {
	int number = 10;
	if (number > 0) {
	System.out.println("Number is positive.");
	}
	System.out.println("This statement is always executed.");
	}
	}
		Number is Positive.
		
		This statement is always executed.

2. Write a program to make the use of logical operators.
3. Write a program to check no is even or odd.

(Space for Answer)

2) public class logical_opertors {
public static void main(String[] args) {
boolean bool1 = true, bool2 = false;
System.out.println("bool1 && bool2 = " + (bool1 && bool2));
System.out.println("bool1 || bool2 = " + (bool1 | bool2));
System.out.println("!(bool1 && bool2) = " + !(bool1 && bool2));
}
}

```
3)  public class EvenOdd {
    public static void main(String[ ] args) {
     int num = 10 ;
     if(num % 2 == 0)
     System.out.println(num + " is even");
      else
     System.out.println(num + " is odd");
      }
   }
```

Practical No. 4: Develop programs to demonstrate use of switch – case statement and conditional if (?:)

I. Practical Significance:

Java uses control statements to control the flow of execution of program based on certain conditions. These are used to cause the flow of execution to advance and branch based on condition. Students will be able to use switch-case to check the multiple conditions.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java"

The practical is expected to develop the following skills:

1. Write a program to use switch-case to check multiple conditions.
2. Develop a program to check condition in one line.

IV. Relevant Course Outcome(s)

Develop programs using Object Oriented methodology in Java.

V. Practical Outcome (PrOs)

Develop programs to demonstrate use of switch - case statement and conditional if (?:)

VI. Relevant Affective domain related Outcome(s)

1. Follow safety practices.
2. Practice good housekeeping
3. Demonstrate working as a leader/ a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

1. Decision making using Switch-case:

Syntax:

```
switch(expression)
{
case value1 :
    // Statements
break; // break is optional
```

```
case value2 :
    // Statements
break; // break is optional
        .
        .
        .
case valueN :
    // Statements
break; // break is optional

default :
    // Statements
}
```

2. Conditional if (ternary operator):

Syntax:

```
result = testStatement ? value1 : value2;
```

VIII. Resources required (Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer (13-15 preferable). RAM minimum 2 GB		
2	Software	jdk1.8.0 or above.		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Write any program using switch-case statement.

Program :

```
public class SwitchCase {
public static void main(String args[]){
    int num=2;
    switch(num+2){
 case 1:
   System.out.println("Case1: Value is: "+num);
  case 2:
   System.out.println("Case2: Value is: "+num);
  case 3:
   System.out.println("Case3: Value is: "+num);
     default:
   System.out.println("Default: Value is: "+num);
    }
  }
}
```

XI. Result (Output of Code):

Default: Value is: 2

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. What will happen if break is not written for a case in switch case?
2. When default case is executed?
3. List datatypes allowed in switch expression?
4. Write a program to make use of ternary operator.

(Space for Answer)

1) If we do not use break statement at the end of each case, program will execute all consecutive case statements until it finds next break statement or till the end of switch case block.

2) The default statement is executed if no case constant-expression value is equal to the value of expression . If there's no default statement, and no case match is found, none of the statements in the switch body get executed.

3) A switch works with the byte , short , char , and int primitive data types.

4) Program :

```
public class Main {
public static void main(String[] args) {
int number = 24;
if(number > 0) {
System.out.println("Positive Number");
}
else {
System.out.println("Negative Number");
    }
  }
}
```

XIII. Exercise:

1. Write Error/output of code in the given space.

Sr. No.	Program Code	Error/Output

1.	public class SwitchCaseExample1 {
	public static void main(String args[]){
	int num=2;
	switch(num+2)
	{
	case 1:
	System.out.println("Case1: Value is: "+num);
	case 2:
	System.out.println("Case2: Value is: "+num);
	case 3:
	System.out.println("Case3: Value is: "+num);
	default:
	System.out.println("Default: Value is: "+num);
	}
	}
	}
		Default: Value is: 2

2.	public class Program {
	public static void main(String[] args) {
	int value = 100;
	switch (value) {
	case 100:
	System.out.println(true);
	break;
	case 100:
	System.out.println(true);
	break;
	}
	}
	}
		error: duplicate case label
		case 100:

2. Write any program to check switch-case statement using character datatype.

(Space for Answer)

.....

.....

.....

.....

.....

.....

.....

.....

2) Program :

```
public static void main(String args[])
{
 public class SwitchCaseCharacterdatatype
{
   char ch='b';
   switch(ch)
{
   case 'd':
   System.out.println("Case1 ");
   break;
   case 'b':
   System.out.println("Case2 ");
    break;
    case 'x':
    System.out.println("Case3 ");
     break;
     case 'y':
     System.out.println("Case4 ");
      break;
     default:
     System.out.println("Default ");
      }
  }
}
```

Practical No. 5: Develop programs to demonstrate use of looping statement 'for'.

I. Practical Significance:

A for loop is used to execute a block of code several times. Students will be able to use for loop to replace the repetition of statements.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java".

The practical is expected to develop the following skills:

1. Develop a program to using for loop

IV. Relevant Course Outcome(s)

Develop programs using Object Oriented methodology in Java.

V. Practical Outcome (PrOs)

Develop programs to demonstrate use of looping statement 'for'

VI. Relevant Affective domain related Outcome(s)

1. Follow safety practices.
2. Practice good housekeeping
3. Demonstrate working as a leader/ a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

Syntax:

```
for (initialization condition; testing condition;  increment/decrement)
{
statement(s);
}
```

VIII. Resources required (Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer i3 RAM 2 GB minimum	As per batch size	
2	Software	JDK 1.8.0		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Develop a program to print command line argument using for loop.

```
public class Main
{
public static void main(String[] args)
{
System.out.println("Command-Line arguments are");
for(String str: args)
 {
System.out.println(str);
 }
 }
}
```


XI. Result (Output of Code):

```
Command-Line arguments are
```

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. When for loop will be terminated?
2. Can we write a for loop without initialization? If yes, give example.
3. Write a for loop to increment index variable by 2 in each iteration.
4. When for loop will be executed infinitely?

(Space for answer)

1) when the break statement is encountered inside a loop, the loop is immediately terminated and the program control resumes at the next statement following the loop. It is also used to exit a loop.

2) Yes, a for loop can be written without initialization.
for loop statement usually goes like: for (initialization; test-condition; update).

3)
```
public class Hello
{
public static void main (string args [])
{
  int value = 0;
  for (int i =0 ; I++)
 {
    if (i %2 ==0)
 {
    value++;
       }
     }
    }
 }
```

4) Hence, a loop which will never stop or i.e over end is named to be infinite loop.

It the terminating condition is not given or if the condition cause the loop to start again then it is called as infinite loop.

XIII. Exercise:

1. Write any program using if condition with for loop.
2. Write any program to display pyramids of stars/patterns using increment/decrement

(Space for Answer)

1)
```
class check
{
public static void main(string args[])
 {
   for (i=1; i<100; i++ )
 {
    if (i%2 ==0)
```

```
        {
        System.out.print (;);
         }
        }
      }
    }
```

2)
```
public class JavaProgram
{
public static void main(String args[])
{
 int i, j;
 for(i=0; i<5; i++)
 {
  for(j=0; j<=i; j++)
 {
  System.out.print("* ");
 }
  System.out.println();
 }
 }
}
```

Practical No. 6: Develop programs to demonstrate use of 'while', 'do-while'

I. Practical Significance:

Loop is used in programming to repeat a specific block of code until certain condition is true. Students will be able to use while and do-while loop to replace the repetition of statements.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java".

The practical is expected to develop the following skills:

1. Develop a program to using while and do-while loop.

IV. Relevant Course Outcome(s)

Develop programs using Object Oriented methodology in Java.

V. Practical Outcome (PrOs)

Develop programs to demonstrate use of 'while', 'do-while'

VI. Relevant Affective domain related Outcome(s)

1. Follow safety practices.
2. Practice good housekeeping
3. Demonstrate working as a leader/ a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

1. While loop:

Syntax:

```
while(condition)
{
  statement(s);
}
```

2. do-while loop:

Syntax:

```
do {
  // Statements
} while (Boolean_expression);
```

VIII. Resources required (Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer i3 RAM minimum 2 GB	As per batch size	
2	Software	jdk1.8.0 or above.		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Develop a program to use logical operators in do-while loop.

```
public class Main {
public static void main(String[] args) {
 int i = 1, n = 5;
   do {
     System.out.println(i);
     i++;
   } while(i <= n);
 }
}
```


XI. Result (Output of Code):

Output :

```
1
2
3
4
5
```

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. State difference between while and do-while loop.
2. In do-while loop termination condition is checked at _____(beginning/end)
3. How many times do-while loop will be executed if condition is false?

(Space for answer)

1)

While	**do-while**
1)While loop first check the condition then enter in the body.	1) do-while loop firstly enter in the loop then check the condition.
2) While loop is entry controlled loop	2) do-while loop is exit controlled loop.
3) In while condition comes before body.	3) In do-while loop condition comes after body.

2) In do-while loop termination condition is checked at **end**

3) As I mentioned in the beginning of this guide that do-while runs at least once even if the condition is false because the condition is evaluated, after the execution of the body of loop.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

XIII. Exercise:

1. Write Error/output of code in the given space.

Sr. No.	Program Code	Error/Output

1.	class DoWhileBasics
	{
	public static void main(String args[])
	{
	int a=1;
	do
	{
	System.out.println(a);
	a=a+1; // or a++;
	}
	while(a<=10);
	}
	}
		1
		2
		3
		4
		5
		6
		7
		8
		9
		10

2.	class Test {
	public static void main(String[] args)
	{
	while(true) {
	System.out.print(1);
	do {
	System.out.print(2);
	} while (false);
	}
	}
	}
		12121212121212
		12121212121212
		12121212121212
		12121212121212
		12121212121212

2. Write a program to display number 1 to 50 using do-while loop.

XIV. References/ Suggestions for Further Reading

1. https://www.codesdope.com/c-loop-and-loop/
2. https://www.youtube.com/watch?v=llX6cLed73o
3. https://www.journaldev.com/16536/java-do-while-loop
4. https://beginnersbook.com/2015/03/do-while-loop-in-java-with-example/

(Space for answer)

2)Program :

```
public class Main {
public static void main(String[] args) {
int i = 1, n = 50;
do {
System.out.println(i);
 i++;
}
while(i <= n);
}
}
```

Practical No. 7 and 8: Develop programs for implementation of implicit type casting in Java, Part –I and Part – II.

I. Practical Significance:

Assigning a value of one type to a variable of different type is known as Type Casting. When you assign value of one data type to another, the two types might/ might not be compatible with each other. Students will be able to understand implicit type conversion between data types.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

Develop Applications using Java.

IV. Relevant Course Outcome(s)

"Develop Applications using Java".

The practical is expected to develop the following skills:

1. Develop a program to show automatic conversion between various compatible data types.

V. Practical Outcome (PrOs)

Develop programs for implementation of implicit type casting in Java.

VI. Relevant Affective domain related Outcome(s)

1. Follow safety practices.
2. Practice good housekeeping
3. Demonstrate working as a leader/ a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

1. Widening or automatic type conversion: Possible when two types are compatible and target type is greater than source type.
2. Narrowing may result in loss of information.

Following table shows the casts that result in a loss of information.

Sr. No.	From	To
1.	byte	short, char, int, long, float, double
2.	short	int, long, float, double
3.	char	int, long, float, double
4.	long	float, double
5.	float	double

VIII. Resources required (Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer i3 Ram 2GB minimum	As per batch size	
2	Software	JDK 1.8.0 or above		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Develop a program to show the use of implicit typecasting.

```
public class ImplicitTypecasting {
public static void main(String args[])
{
    byte p = 12;
    System.out.println("byte value : "+p);
    short q = p;
    System.out.println("short value : "+q);
    int r = q;
    System.out.println("int value : "+r);
    long s = r;
    System.out.println("long value : "+s);
    float t = s;
    System.out.println("float value : "+t);
    double u = t;
    System.out.println("double value : "+u);
  }
}
```

XI. Result (Output of Code):

```
byte value : 12
short value : 12
int value : 12
long value : 12
float value : 12.0
double value : 12.0
```

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. List different data types according to storage capacity.
2. State need of typecasting.
3. State the data types to which boolean datatype is implicitly casted.
4. Write two examples of implicit type casting.

(Space for answer)

1) 1) Byte.
2) Short.
3) Int.
4) Long.
5) Float.

2) Typecasting, is a method of changing an entity from one data type to another.
It is used to ensure variables are correctly processed by a function.

3) A boolean data type represents only one bit of information either true or false, but the size of the boolean data type is virtual machine-dependent.
Values of type boolean are not converted implicitly or explicitly (with casts) to any other type.

4) 1)
```
public class ImplicitType
{
 public static void main (String[] args)
{
  byte  i=10;
  long  j=1+8;
 double k=i*2.5 +j;
 System.out.println("I =" +i);
 System.out.println("J =" +j);
 System.out.println("K =" +k);
  }
 }
```

4) 2)
```
public class ImplicitTypecastingExample {
 public static void main(String args[]) {
  byte p = 12;
 System.out.println("byte value : "+p);
  short q = p;
 System.out.println("short value : "+q);
  int r = q;
 System.out.println("int value : "+r);
  long s = r;
  System.out.println("long value : "+s);
  float t = s;
  System.out.println("float value : "+t);
   double u = t;
   System.out.println("double value : "+u);
     }
  }
```

XIII. Exercise:

1. Write Error/output of code in the given space.

Sr. No.	Program Code	Error/Output

1.	class Test{
	public static void main(String[] args) {
	int i = 100;
	long l = i;
	float f = l;
	System.out.println("Int value "+i);
	System.out.println("Long value "+l);
	System.out.println("Float value "+f);
	}
	}
		Output :
		
		Int value 100
		Long value 1
		Float value 1.0

2.	public class Test{
	public static void main(String[] argv) {
	char ch = 'c';
	int num = 88;
	ch = num;
	}
	}
		Error:
		possible lossy conversion from int to char
		ch = num;

3. Write a program to implicitly typecast lower range data type to larger storage size datatype.


```
public class Test
{
public static void main(String[] args)
 {
  int i = 100;
  long l = i;
  float f = l;
  System.out.println("Int value "+i);
  System.out.println("Long value "+l);
  System.out.println("Float value "+f);
 }
}
```

Practical No. 9: Develop programs for implementation of explicit type conversion in Java.

I. Practical Significance:

Assigning a value of one datatype to another datatype is known as typecasting. When larger type of data should be assigned to smaller datatype explicit typecasting is required. Students will be able to implement assignment of explicit typecasting.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java".

The practical is expected to develop the following skills:

1. Develop a program to show use of explicit type casting.

IV. Relevant Course Outcome(s)

Develop programs using Object Oriented methodology in Java.

V. Practical Outcome (PrOs)

Develop programs for implementation of explicit type conversion in Java.

VI. Relevant Affective domain related Outcome(s)

1. Follow safety practices.
2. Practice good housekeeping
3. Demonstrate working as a leader/ a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

Narrowing or Explicit Conversion

If we want to assign a value of larger data type to a smaller data type we perform type casting explicitly or narrowing.

- This is useful for incompatible data types.
- Here, target-type specifies the desired type to convert the specified value to.

Syntax:

dataType variableName = (dataType) variableToConvert;

Example:

```
float a =5.2;
int b = (float) a;
```

VIII. Resources required (Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer i3 2GB Ram minimum	as per batch size	
2	Software	Jdk 1.8.0 or above		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Develop a program to show the use of explicit type casting.

```
public class ExplicitProgram
{
   public static void main(String[] args)
{
      int a=257;
      byte b =(byte)a;
      System.out.print(b);
}
}
```


XI. Result (Output of Code):

.....

Output = 1

.....

.....

.....

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. What is casting?
2. What is difference between implicit and explicit type casting?
3. What is narrowing?

(Space for answer)

1) Type casting is the process of converting an entity of one data type into another data type.

2)

Implicit	Explicit
1) implicit means done by Java virtual machine (JVM)	1) Explicit means done by programer.
2) This type of conversion happens automatically by the compiler	2) In this type of conversion the program deliberately converts the answer or variable in a specific data type of his choice

3) The narrowing conversion occurs form a data type to a different type that has smaller size such as form a long (64 bits) to an (32 bits) In general, narrowing primitive conversion can occur in these case short to byte or char.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

XIII. Exercise:

1. Write Error/output of code in the given space.

Sr. No.	Program Code	Error/Output

1.	class Test{
	public static void main(String[] args) {
	double d = 100.04;
	long l = (long)d;
	int i = (int)l;
	System.out.println("Double value "+d);
	System.out.println("Long value "+l);
	System.out.println("Int value "+i);
	}
	}
		Double value 100.04
		Long value 100
		Int value 100

2.	class Test{
	public static void main(String args[]) {
	byte b = 50;
	b = (byte)(b * 2);
	System.out.println(b);
	}
	}
		100

3.	class Test{
	public static void main(String args[]) {
	byte a= 4;
	char b = 'z';
	short c = 102;
	int i = 5000;
	float f = 5.7f;
	double d = .124;
	double result = (f * a) + (i / b) - (d * c);
	System.out.println("result = " + result);
	}
	}
		result = 50.151999

2. Write a program to convert variable of basic datatypes and shows result of explicit typecasting.

(Space for Answer)

Program :

```
public class Explicit
{
 public static void main(String[] args)
{
  byte x = 10;
  long y = (long)x;
  double z = (double)y;
  double result = x+y+z;
 System.out.println(result);
  }
 }
```

Practical No. 10: Develop program for implementation of constructor and multiple constructors.

I. Practical Significance:

Constructor in Java or any programming language is a special type of method that is used to initialize the object. Different types of constructors are used to initialize the object in different way. The student will be able to use different types of constructor for creating the object.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitation
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java".

The practical is expected to develop the following skills:

1. Write a program to initialized objects using constructor.
2. Develop a program to pass message among objects.

IV. Relevant Course Outcome(s)

Develop programs using Object Oriented methodology in Java.

V. Practical Outcome (PrOs)

Develop programs for implementation of single constructor and multiple constructors.

VI. Relevant Affective domain related Outcome(s)

1. Follow safety practices.
2. Practice good housekeeping
3. Demonstrate working as a leader/ a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

Types of constructors

1. Default constructor (No-argument constructor)-
   General Syntax –
   class_name()
   {
   Statements for initialize the data members.
   }

2. Parameterized constructor
   General Syntax –
   class_name(parameter list)

{
Statements for initialize the data members.
}

3. Copy Constructor
General Syntax –
class_name (classname reference)
{
Statements for initialize the data members.
}

VIII. Resources required(Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	computer systeam	Computer i3 2GB Ram minimum	As per batch size	
2	Software	JDK 1.8.0 or above		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Demonstrate use of at least two types of constructors.

```
class Example
{
    private int var;
    public Example()
    {
        this.var = 10;
    }
    public Example(int num)
    {
        this.var = num;
    }
    public int getValue()
    {
         return var;
    }
    public static void main(String args[])
    {
         Example obj = new Example();
         Example obj2 = new Example(100);
         System.out.println("var is: "+obj.getValue());
         System.out.println("var is: "+obj2.getValue());
    }
}
```

XI. Result (Output of Code):

var is: 10
var is: 100

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Does constructor return a value?
2. Specify the situation when the default constructor is provided by the system.
3. Specify the situation when the default constructor is explicitly defined in the class.
4. How constructor overloading can be done?

(Space for answer)

1) No, constructor does not return a value. It returns the current class instance but you cannot use return type yet it returns a value.

2) A default constructor refers to a nullary constructor that is automatically generated by the compiler if no constructors have been defined for the class.
The default constructor implicitly calls the superclass's nullary constructor, then executes an empty body.

3) If constructors are explicitly defined for a class, but they are all non-default, the compiler will not implicitly define a default constructor, leading to a situation where the class does not have a default constructor.

4) The constructor overloading can be defined as the concept of having more than one constructor with different parameters so that every constructor can perform a different task.

XIII. Exercise:

1. Write output of code in the given space.

Sr. No.	Program Code	Output

1.	class T {
	int t;
	}
	class Main {
	public static void main(String args[])
	{
	T t1 = new T();
	System.out.println(t1.t);
	}
	}
		0 (zero)

2. Modify the following program to execute without error. State which constructors are used in the program.

```
class Point
{
   int m_x, m_y;
   public Point(int x, int y)
   {   m_x = x;   m_y = y;  }
 public static void main(String args[])
 {
      Point p1 = new Point();
      Point p = new Point(2,3);
     System.out.println("X"+p.m_x);
     System.out.println("Y"+p.m_y);
     System.out.println("X"+p1.m_x);
     System.out.println("Y"+p1.m_y);
  }
}
```


3. Write a program to implement different types of constructors to perform addition of complex numbers.

(Space for Answer)

.....

.....

.....

.....

.....

.....

.....

.....

.....

3)

```
 public class Complex
{
int Real,Imag;
Complex()
{}
Complex(int Real1,int Imag1)
{
Real=Real1;
Imag=Imag1;
}
Complex AddComplex(Complex C1,Complex C2)
{
Complex CSum=new Complex();
CSum.Real=C1.Real+C2.Real;
CSum.Imag=C1.Imag+C2.Imag;
return CSum;
}
}

class Complexmain
{
public static void main(String[] a)
{
Complex C1=new Complex(4,8);
Complex C2=new Complex(5,7);
Complex C3=new Complex();
C3=C3.AddComplex(C1,C2);
System.out.println("SUM:" + C3.Real +"+i" + C3.Imag);
}
}
```

Practical No. 11 and 12: Develop program for implementation of different functions of String Class, Part – I and Part – II.

I. Practical Significance:

String is a sequence of characters. Java String is a powerful concept. Students will be able to perform various operations on String object using different methods of String class.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java".

The practical is expected to develop the following skills:

1. Write a program to initialized objects using constructor.
2. Develop a program to pass message among objects.

IV. Relevant Course Outcome(s)

Develop programs using Object Oriented methodology in Java.

V. Practical Outcome (PrOs)

Develop program for implementation of different functions of String Class.

VI. Relevant Affective domain related Outcome(s)

1. Follow safety practices.
2. Practice good housekeeping
3. Demonstrate working as a leader/ a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

String class methods:

Sr. No.	Syntax	Task Performed
1.	public char charAt(int index)	Returns character value at specified index
2.	public int compareTo(String anotherString)	Compares two strings and returns int
3.	public boolean equals(Object anotherObject)	Compares two strings.
4.	public boolean equalsIgnoreCase (String str)	Compares two strings, ignoring cases.

5.	public int length()	Returns length of string
6.	public String replace(char oldChar, char newChar)	Returns a string replacing all the old char or CharSequence to new char or CharSequence.
7.	public boolean startsWith(String prefix)	Checks if this string starts with given prefix.
8.	public boolean endsWith(String suffix)	Checks if this string ends with given suffix.
9.	int indexOf(int ch)	A method returns index of given character value or substring.
10.	String substring(int startIndex)	Returns new string that is substring of this string
11.	int lastIndexOf(int ch)	A method returns last occurrence of given character value or substring.

String Buffer class methods:

Sr. No.	Syntax	Task Performed
1.	StringBuffer append (StringBuffer sb)	Appends specified StringBuffer with this StringBuffer
2.	StringBuffer insert (int offset, String str)	This method inserts a string str at position mentioned by offset.
3.	void setLength(int newlength)	Sets the length of the character sequence
4.	void setCharAt(int index, char ch)	The character at specified index of this StringBuffer is set to ch.
5.	StringBuffer reverse()	Reverse the character sequence in this StringBuffer.

VIII. Resources required(Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB	As per batch size	
2	Software	JDK 16.0.1		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Write a program to show the use of all methods of String class.

```
public class Example
{
  public static void main(String args[])
{
 String str = "Beginnersbook";
 char arrch[]={'h','e','l','l','o'};
 String str2 = new String(arrch);
 String str3 = new String("Java String Example");
 System.out.println(str);
 System.out.println(str2);
 System.out.println(str3);
  }
}
```

XI. Result (Output of Code):

Beginnersbook
hello
Java String Example

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. List different constructors of String class along with syntax
2. List different constructors of StringBuffer class along with syntax.
3. State whether String is primitive datatype or class in Java? State the package.
4. What is difference between = = , equals() and compareTo() method?

(Space for answer)

1. 1) String()
   2) String(byte[ ] byte)
   3) String (Char [ ] value)
   4) String (String original)

2. StringBuffer s=new StringBuffer();
   StringBuffer insert(int index, String str)
   void ensureCapacity(int capacity)
   public StringBuffer appendCodePoint(int codePoint)
   public char charAt(int index)

3. String such as converting string instance and performing string concatenation, String is not a primitive type, but a class.
   String is a class in Java and reference data type.
   String is present in Java. Lang package.

4. equal ( )	Compare ( )
1) This method is used to compare two strings.	1) This method compare the string using directory order.
2). If two contents of string 'Str' is same as invoking string then it returns true otherwise returns false.	2). A String is less than another if it comes before the other in dictionary order.

XIII. Exercise:

1. Write output of code in the given space.

Sr. No.	Program Code	Output

1.	class String_demo{
	public static void main(String args[]) {
	char chars[] = {'a','b','c'};
	String s = new String(chars);
	System.out.println(s);
	}
	}
		Output :
		abc

2.	class Output{
	public static void main(String args[]) {
	String s1 = "Hello I love Java" ;
	String s2 = new String(s1);
	System.out.println((s1= = s2) + " " + s1.equals(s2));
	}
	}
		False
		True

2. Write a program to implement all methods of StringBuffer class.

(Space for Answer)

2.
```
class Str
{
public static void main( String args[] )
{
StringBuffer s = new StringBuffer("Coding Atharva");
System.out.println("\n String = "+s);
System.out.println("\n Length = "+s.length() );
System.out.println("\n Length = "+s.capacity() );
 s.setLength(6);
System.out.println("\n After setting length String = "+s );
 s.setCharAt(0,'K');
System.out.println("\n SetCharAt String = "+s );
 s.setCharAt(0,'C');
 int a = 007;
s.append(a);
System.out.println("\n Appended String = "+s );
s.insert(6," Atharva");
System.out.println("\n Inserted String = "+s );
s.reverse();
System.out.println("\n Reverse String = "+s );
s.reverse();
s.delete(6,14);
System.out.println("\n\n After deleting string="+s);
}
}
```

Practical No. 13: Develop program for implementation of Arrays in Java

I. Practical Significance:

Arrays store homogeneous data i.e same type of data in consecutive memory locations which will help to fetch data in constant access time. Students will be able to use array to refer multiple values with same name.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and Teamwork:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java".

The practical is expected to develop the following skills:

1. Develop a program to perform various operations using array.

IV. Relevant Course Outcome(s)

Develop programs using Object Oriented methodology in Java.

V. Practical Outcome (PrOs)

Develop program for implementation of Arrays in Java

VI. Relevant Affective domain related Outcome(s)

1. Follow safety practices.
2. Practice good housekeeping
3. Demonstrate working as a leader/ a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

1. Creating an array:

```
dataType[] arrayRefVar;
or
dataType arrayRefVar[];
arrayRefVar = new type [size];
```

2. Array of Objects:

```
Class_name array_name = new class_name[size];
```

3. Types of arrays:

1. One Dimensional

   Example:

```
int [] intArray = new int[20];
```

2. Multi Dimensional
   Example:
   int[][] intArray = new int[10][20]; //a 2D array or matrix
   int[][][] intArray = new int[10][20][10]; //a 3D array

VIII. Resources required(Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer (i3-i5 preferable), RAM	As per batch size	
2	Software	minimum 2 GB Java J.D.K 1.16.0		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Write a program to implement multidimensional array.


```
class Testarray
{
public static void main(String args[])
{
int a[]=new int[5];
a[0]=10;//initialization
a[1]=20;
a[2]=70;
a[3]=40;
a[4]=50;
for(int i=0;i<a.length;i++)
System.out.println(a[i]);
}
}
```

XI. Result (Output of Code):

10
20
70
40
50

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. What is use of new operator in defining an array?
2. In 2D array which dimension is optional at the declaration of array?
3. Is it possible to change size of array once allocated?
4. State the situation where Index Out of Bounds Exception will be generated.

(Space for answer)

1. An array is an instance of a special Java array class and has a corresponding type in the type system.
This means that to use an array, as with other objects, we first declare a variable.

2. In 2D array dimension the second dimension is optional.

3. In Java array cannot be resized onee it is declared.
If you want a dynamic data structure with random access you use array.

4. If a request for a negative or an index greater than or equal to size of array is made, then the Java throws a array / Index out of Bounds Exception.

XIII. Exercise:

1. Write output/error of code in the given space.

Sr. No.	Program Code	Output/Error

1.	State line no and error.
	classTest2 {
	public static void main(String[] args) {
	inta[] = new int[5]; // line 1
	int[] arr = new int[]; // line 2
	}
	}
		4 error

2.	class Test5 {
	public static void main(String[] args) {
	int arr[] = new int[5];
	System.out.println(arr);
	System.out.println(arr[0]);
	}
	}
		[I@2c7b84de0

3. Write a program to display array elements using for-each loop.

(Space for answer)

```
class ForEach
{
 public static void main( String args[] )
 {
  int a[ ]={1,1,2,3,4,5,5};

  for(int i: a  )
  {
   System.out.println(i);
  }
  }
}
```

Practical No. 14: Develop program for implementation of Vectors in Java

I. Practical Significance:

Vector implements a dynamic array. Vector hold different number of objects. Students will be able to use Vectors in the program efficiently.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java".

The practical is expected to develop the following skills:

1. Develop a program to perform various operations on Vector using different methods.

IV. Relevant Course Outcome(s)

Develop programs using Object Oriented methodology in Java.

V. Practical Outcome (PrOs)

Develop program for implementation of Vectors in Java

VI. Relevant Affective domain related Outcome(s)

1. Follow safety practices.
2. Practice good housekeeping
3. Demonstrate working as a leader/ a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

Vector class methods:

Sr. No.	Syntax	Task Performed
1.	void addElement(Object ob)	Adds the specified component to the end of this vector increasing its size by one
2.	int capacity()	Returns the current capacity of this vector
3.	boolean contains(Object elem)	Tests if the specified object is a component in this vector.
4.	void clear()	Removes all the elements from this vector
5.	Object elementAt(int index)	Returns the component at specified index

6.	Enumeration elements()	Returns enumeration of components of this vector
7.	Object firstElement()	Returns first component of this vector
8.	Object lastElement()	Returns last component of this vector
9.	int indexOf(Object elem)	Searches for the first occurrence of given argument.
10.	void insertElementAt(Object obj, int index)	Inserts specified object as a component at the specified index position.
11.	void removeElementAt(int index)	Removes the element at specified position in the vector
12.	boolean removeElement(Object obj)	Removes first occurrence of the argument from the vector
13.	int size()	Returns number of components in the vector
14.	void copyInto(Object[] array)	Copies the components of vector into specified array.

VIII. Resources required(Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer i3 RAM minimum 2 GB	As per batch size	
2	Software	jdk1.8.0 or above.		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Write a program to insert different elements in the Vector and display them.

```
import java.util.Vector;
class Main {
  public static void main(String[] args) {
    Vector<String> mammals= new Vector<>();
    mammals.add("Dog");
    mammals.add("Horse");
    mammals.add(2, "Cat");
    System.out.println("Vector: " + mammals);
    Vector<String> animals = new Vector<>();
    animals.add("Crocodile");
    animals.addAll(mammals);
    System.out.println("New Vector: " + animals);
  }
}
```

XI. Result (Output of Code):

Vector: [Dog, Horse, Cat]
New Vector: [Crocodile, Dog, Horse, Cat]

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. State difference between size() and capacity() method of Vector class.
2. Differentiate between addElement() and insertElementAt() methods of Vector class.
3. Differentiate between array and Vector.

(Space for answer)

1. Size ( )	Capacity ( )
1) Size( ) returns the number of element currently in the vector.	1). Capacity( ) is the return capacity of the vector.

2. addElement( )	insertElement ( )
The object specificed by element is added to the end of vector.	Add the element to the vector at the location specified index.

3. Array	Vector
1) Array is fixed in Size.	1) Vector is dynamic in size.
2). It holds value of primitive and object type.	2). It holds only object type Value.
3) It is not class not contain any method.	3) Vector is a class and contain lots of method .

XIII. Exercise:

Sr. No.	Program Code	Output

1.	import java.util.*;
	class demo1 {
	public static void main(String[] args)
	{
	Vector v = new Vector(20);
	System.out.println(v.capacity());
	System.out.println(v.size());
	}
	}
		20
		0

1. Write a program to use different methods of Vector class.

(Space for answer)

1. Write a program to use different methods of vector class.

```
import java.util.Vector;
public class VectFun
{
 public static void main( String args[] )
 {
  Vector v = new Vector();
  v.addElement(" Coding with Raman ");
  v.add(007);
 System.out.println(" Capacity= "+v.capacity() );
 System.out.println(" Contains= "+v.contains(007) );
 System.out.println(" Element At 0= "+v.elementAt(0) );
 System.out.println(" Element At 0= "+v.elementAt(0) );
 System.out.println(" Enumeration of Components = "+v.elements() );
 System.out.println(" First Element = "+v.firstElement() );
 System.out.println(" Last Elements = "+v.lastElement() );
 System.out.println(" Index Of = "+v.indexOf(007) );
 v.insertElementAt(1,0);
 System.out.println(" Elements After Inserting = "+v );
 v.removeElementAt(0);
 System.out.println(" Elements After Removing = "+v );
 System.out.println(" Remove Element = "+v.removeElement(007) );
 System.out.println(" Elements After Removing = "+v );
 System.out.println(" Size = "+v.size() );
 Object obj[] = new Object[10];
 v.copyInto(obj);
 v.clear();
 System.out.println(" Elements After Clearing = "+v );
   }
 }
```

Practical No. 15 and 16: Develop a program for implementation of Wrapper Class to convert primitive into object and object into primitive.

I. Practical Significance:

Wrapper classes are used to convert primitive data type into an object. The primitive data types are not objects. They are predefined in the language itself. Student should be able to use different wrapper classes and their methods.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and Team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java".

The practical is expected to develop the following skills:

1. Develop a program to create object of primitive data types and use them.

IV. Relevant Course Outcome(s)

Develop programs using Object Oriented methodology in Java.

V. Practical Outcome (PrOs)

Develop a program for implementation of Wrapper Class to convert primitive into object.

VI. Relevant Affective domain related Outcome(s)

1. Follow safety practices.
2. Practice good housekeeping
3. Demonstrate working as a leader/ a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

Integer class methods:

Sr. No.	Method	Purpose
1.	parseInt(s)	returns a signed decimal integer value equivalent to string s
2.	toString(i)	returns a new String object representing the integer i
3	byteValue()	returns the value of this Integer as a byte
4	doubleValue()	returns the value of this Integer as a double
5	floatValue()	returns the value of this Integer as a float

6	intValue()	returns the value of this Integer as an int
7	shortValue()	returns the value of this Integer as a short
8	longValue()	returns the value of this Integer as a long
9	int compareTo(int i)	Compares the numerical value of the invoking object with that of i. Returns 0 if the values are equal. Returns a negative value if the invoking object has a lower value. Returns a positive value if the invoking object has a greater value.
10	static int compare(int num1, int num2)	Compares the values of num1 and num2. Returns 0 if the values are equal. Returns a negative value if num1 is less than num2. Returns a positive value if num1 is greater than num2.
11	boolean equals(Object intObj)	Returns true if the invoking Integer object is equivalent to intObj. Otherwise, it returns false.

Similar Wrapper class methods are available for Float, Short, Long and Double class.

VIII. Resources required(Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer i3 RAM minimum 2GB	As per batch size	
2	Software	jdk1.8.0 or above.		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Write a program to show the use of Integer Wrapper class methods.

```
public class WrapperExample3{
public static void main(String args[]){
int i=30;
Integer intobj=i;
System.out.println("Integer object: "+intobj);
}
}
```

XI. Result (Output of Code):

30

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Write a different ways to create object of the any primitive datatype.
2. Write methods of Number class to convert object into primitive datatypes.
3. List all Wrapper classes in Java.

(Space for answer)

1. byte byte value ( )
   double double value ( )
   Float Float value ( )
   int int value
   long long value
   Short Short value

2. a. Integer : Return object value as integer.
   b. Byte : Return object value as byte.
   c. Double : Return object value as double.
   d. Short : Return object value as short.
   e. Float : Return object value as float.
   f. Long : Return object value as long.

3. list of wrapper class in Java :
   1.Character
   2.Byte
   3.Short
   4.Integer
   5.Long
   6.Float
   7.Double
   8.Boolean

XIII. Exercise:

1. Write a program to convert String value into Integer Wrapper class object.
2. Write a program to make use of Character Wrapper class methods.
3. Write a program to convert Integer object value into primitive datatype byte, short and double value.

(Space for answer)

1.
```
public class StringToIntExample1 {
public static void main(String args[]) {
String s="200";
int i=Integer.parseInt(s);
System.out.println(i);
}}
```

2.
```
public class charToCharacter {
public static void main(String[] args) {
   char c = 'a';
   Character cObj = new Character(c);
   System.out.println(cObj);
 }
 }
```

Practical No. 17: Develop a program which implements the concept of overriding.

I. Practical Significance:

Method overriding is used to change definition of method in subclass. Runtime Polymorphism is achieved through Method Overriding. super keyword is used to call the parent class method or constructor.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java."

The practical is expected to develop the following skills:

1. Develop a program using super and sub class to override the methods.
2. Develop a program using super keyword to override the methods to achieve the runtime polymorphism.

IV. Relevant Course Outcome(s)

Apply concept of inheritance for code reusability.

V. Practical Outcome (PrOs)

Develop a program which implements the concept of overriding.

VI. Relevant Affective domain related Outcome(s)

1. Follow Safety Practices
2. Practice good housekeeping
3. Demonstrate Working as a leader/a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

Method Overriding in Java

Declaring a method in a subclass (child class) which is already existing in the parent class refers to method overriding in java.

The exact implementation in the subclass (child class) overrides (replaces) the implementation in the superclass by providing a method that has same name, same parameters or signature and same return type as that of method in the super(parent) class.

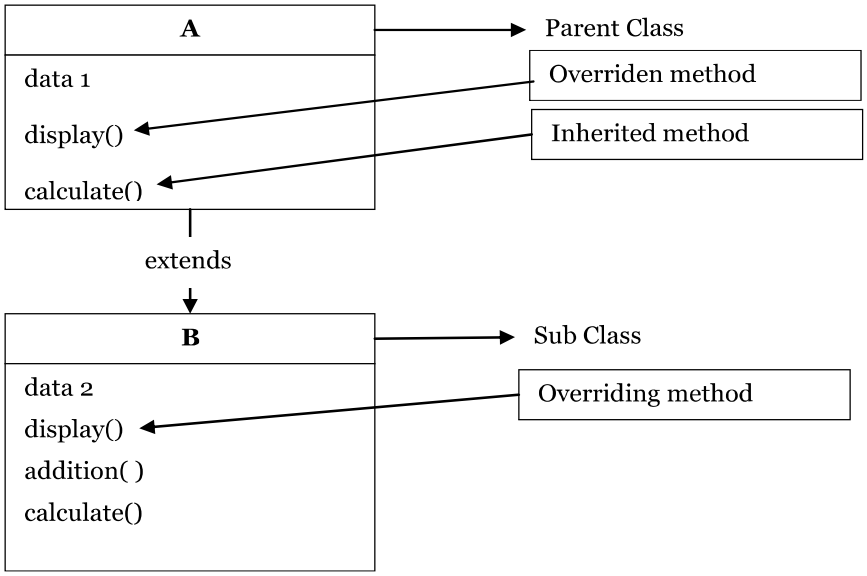


Fig. 5 Overriding of Methods

VIII. Resources required (Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer i3 RAM minimum 2 GB	As per batch size	
2	Software	jdk1.8.0 or above.		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Demonstrate the use of Overriding method display( ) using Super and Sub classes

Note: Attach the code at the end.

```
class MyBaseClass{
  protected void disp()
  {
     System.out.println("Parent class method");
  }
}
class MyChildClass extends MyBaseClass{
  public void disp(){
    System.out.println("Child class method");
  }
  public static void main( String args[]) {
    MyChildClass obj = new MyChildClass();
    obj.disp();
  }
}
```


XI. Result (Output of Code):

Child class method

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. State the difference between method overloading and method overriding.
2. Method Overriding is an example of……(Compile Time Polymorphism/Run Time Polymorphism)
3. Write the rules of method overriding.
4. Write the use of super keyword in method overriding.

(Space for answer)

1. Method overloading	Method overriding
1.Method overloading is a compile time polymorphism.	1. Method overriding is a Run time polymorphism.
2. It is occur within the class.	2. It is performed in two classes with inheritance.

3. Method overloading may or may not require inheritance.	3. While method overriding always needs inheritance.
4. In this, methods must have same name and different signature.	4. While In this, methods must have same name and same signature.

2. Method Overriding is an example of Run Time Polymorphism.

3. The Rules of Method Overriding :
   1. The argument list should be exactly the same as that of the overridden method.
   2. A method declared final cannot be overridden.
   3. A method declared static cannot be overridden but can be re-declared.
   4. If a method cannot be inherited, then it cannot be overridden.
   5. Constructors cannot be overridden.

4. The use of super keyword in method overriding :
   1. To call methods of the superclass that is overridden in the subclass.
   2. To access attributes (fields) of the superclass if both superclass and subclass have attributes with the same name.
   3. To explicitly call superclass no-arg (default) or parameterized constructor from the subclass constructor.

XIII. Exercise:

1. Develop a program to display the rate of interest of banks by method overriding method.
2. Write the output of the following:

```
class MyBaseClass
{
  protected void disp()
  {
     System.out.println("Parent class method");
  }}
class MyChildClass extends MyBaseClass
{
  public void disp()
  {
    System.out.println("Child class method");
  }
  public static void main (String args[])
  {
    MyChildClass obj = new MyChildClass();
    obj.disp();  }}
```

3. Develop a program to extend 'dog' from 'animal' to override 'move( )' method using super keyword.

(Space for answer)

1.
```
public class Bank{
int getRateOfInterest(){return 0;}}
class SBI extends Bank{
int getRateOfInterest(){return 9;}}
class ICICI extends Bank{
int getRateOfInterest(){return 10;}}
class AXIS extends Bank{
int getRateOfInterest(){return 11;}}
class MethodOverriding{
public static void main(String args[]){
SBI s=new SBI();
ICICI i=new ICICI();
AXIS a=new AXIS();
System.out.println("SBI Rate of Interest: "+s.getRateOfInterest());
System.out.println("ICICI Rate of Interest: "+i.getRateOfInterest());
System.out.println("AXIS Rate of Interest:
"+a.getRateOfInterest());}}
```

Practical No. 18: Develop a program for implementation of Single and Multilevel inheritance.

I. Practical Significance:

Inheritance helps to reuse the existing class properties to derive a new class with additional properties. It helps to reduce the memory space, time, frustration and increases the reliability of code. Different types of Inheritance are used to extend the classes in different ways. The student will be able to use different types of inheritance.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java."

The practical is expected to develop the following skills:

1. Write simple programs to extend a subclass from super class
2. Develop programs to extend multipath inheritance.

IV. Relevant Course Outcome(s)

Apply concept of inheritance for code reusability.

V. Practical Outcome (PrOs)

Develop a program for implementation of Single and Multilevel inheritance.

VI. Relevant Affective domain related Outcome(s)

1. Follow Safety Practices
2. Practice good housekeeping
3. Demonstrate Working as a leader/a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

Inheritance:

The process of deriving a **new class** from an **old class** is called as **Inheritance**.

Types of Inheritance:

1. Single Inheritance
2. Multiple Inheritance
3. Hierarchical Inheritance
4. Multilevel Inheritance

Java does not directly implement multiple inheritance, it is implemented using a secondary inheritance path in the form of **interfaces**.

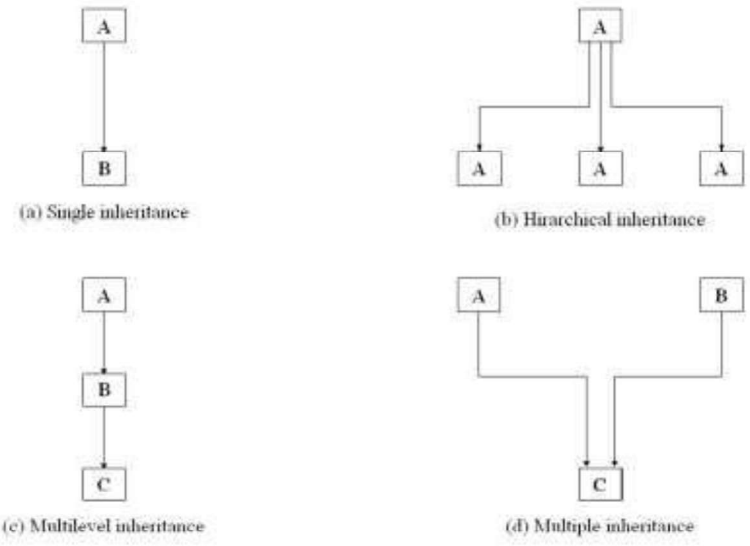


Fig. 6 Forms of Inheritance

Defining a subclass:

```
Class subclassname extends superclassname
{
        variables declaration;
        methods declaration;
}
```

The keyword extends indicates that the properties of the superclassname are extended to the subclassname.

VIII. Resources required(Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer i3 RAM minimum 2 GB	As per batch size	
2	Software	jdk1.8.0 or above.		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Write a program to implement single and multilevel inheritance.

Note: Attach the code at the end.

```
interface A
{
   public void dis(int num);
}

interface B
{
   public void disp(int num);
}
class Mult implements A, B
{
   public void dis(int num)
   {
     System.out.println("From Interface A number is " + num);
   }
   public void disp(int num)
   {
     System.out.println("From Interface B number is " + num);
   }}
public class Test extends Mult
{
   public static void main(String args[])
   {
     Mult ml = new Mult();
     ml.dis(10);
     ml.disp(20);}}
```


XI. Result (Output of Code):

```
From Interface A number is 10
From Interface B number is 20
```

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Justify: Java does not support multiple inheritance.
2. Specify the conditions when the super keyword can be used.
3. Write the importance of final variables and methods.
4. Specify the conditions which needs to be satisfied while using the abstract classes.

(Space for answer)

1. The reason behind this is to prevent ambiguity. Consider a case where class B extends class A and Class C and both class A and C have the same method display(). Now java compiler cannot decide, which display method it should inherit. To prevent such situation, multiple inheritances is not allowed in java.

2. It is end inside subclass method definition. To call a method a defined in super class driver method of superclass is called online public and protected method can called by super keyword it is also and class constructor. Toward constructor its part class super keyword are not used in static method.

3. A final variable can be explicitly initialized only once. A reference variable declared final can never be reassigned to refer to a different object.
   A final class can not be inheritable in Java.

   A final method cannot be overridden by any subclasses. As mentioned previously, the final modifier prevents a method from being modified in a subclass.
   A final method can not be overridden in Java.

4. 1) Abstract method has no body.
2) Always end the declaration with a semicolon(;).
3) It must be overridden. An abstract class must be extended and in a same way abstract method must be overridden.
4) A class has to be declared abstract to have abstract methods.

XIII. Exercise:

1 Develop a program to implement the multilevel inheritance.

2 Develop a program to calculate the room area and volume to illustrate the concept of single inheritance (Assume suitable data wherever necessary).

(Space for answer)

1.
```
class Shape {
    public void display() {
        System.out.println("Inside display");
    }
}
class Rectangle extends Shape {
    public void area() {
        System.out.println("Inside area");
    }
}
class Cube extends Rectangle {
    public void volume() {
        System.out.println("Inside volume");
    }
}
public class Tester {
    public static void main(String[] arguments) {
        Cube cube = new Cube();
        cube.display();
        cube.area();
        cube.volume();
    }
}
```

2.
```
class Room
{
        int length;
        int breadth;
        Room(int x,int y)
        {
                length=x;
                breadth=y;
        }
        int area()
        {
                return(length*breadth);
        }}
class BedRoom extends Room
{
        int height;
        BedRoom(int x,int y,int z)
        {
                super(x,y);
                height=z;
        }
        int volume()
        {
                return (length*breadth*height);
        }}
class InherTest
{
        public static void main(String args[])
        {
                BedRoom room=new BedRoom(14,12,10);
                int area=room.area();
                int volume=room.volume();
                System.out.println("Area = "+area);
                System.out.println("Volume = "+volume);
        }}
```

Practical No. 19: Develop a program for implementation of multiple inheritance.

I. Practical Significance:

Deriving a new class from multiple super/parent classes is known as multiple inheritance. Java doesn't allow multiple inheritance to avoid the ambiguity caused by the overriding methods of the classes. Interface is used to achieve multiple inheritance in java.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java."

The Practical is expected to develop the following skills:

1. Write a program to define and extend the interfaces.
2. Develop a program to implement various forms of interfaces.

IV. Relevant Course Outcome(s)

Apply concept of inheritance for code reusability.

V. Practical Outcome (PrOs)

Develop a program for implementation of multiple inheritance.

VI. Relevant Affective domain related Outcome(s)

1. Follow Safety Practices
2. Practice good housekeeping
3. Demonstrate Working as a leader/a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

1. Interfaces

Java does not support multiple inheritance.

Classes in java cannot have more than one superclass.

A large number of real-time applications require use of multiple inheritance where no. of methods are inherited from several classes. Multiple inheritance proves difficult and adds complexity to the language. Java provides an alternative approach called as **interfaces** to support to the concept of multiple inheritance.

Defining Interfaces

An interface is like a class. Interfaces also contain methods and variables but with a major difference. The difference is that interfaces define only **abstract methods and final fields.**

Interfaces do not specify any code to implement these methods and data fields contain only constants.

Syntax:

```
interface InterfaceName
{
        variables declaration;
        methods declaration;
}
```

Where interface is the keyword and **InterfaceName** is any valid java variable.
Variables are declared as:
static final type VariableName=Value;

Example: return_type methodName1(parameter_list);

Interface Definition:

```
Interface Item
{
        static final int id=100;
        static final String name=”ABC”;
        void display ();
}
```


Extending Interfaces

Interfaces can be extended. The new subinterface will inherit all the members of the superinterface.

```
interface name2 extends name1
{
        body of name2
}
```

Example:

```
interface ItemConstants
{
        int id=100;
        string name=”ABC”;
}

interface Item extends ItemConstants
{
        void display( );
}
```

All constants in one interface

All methods in other interface

2. Multiple interfaces:

```
interface ItemConstants
{
        int id=100;
        string name=”ABC”;
}

interface ItemMethods
{
    void display( );
}

interface Item extends ItemConstants, ItemMethods
{
        ……
        ……
}
```

3. Implementing Interfaces:

Interfaces are used like superclasses whose features/properties are inherited by classes.

It is mandatory to define a class that inherits the given interface.

class classname **implements** interfacename

```
{
        body of classname
}
```

The class classname “implements” the interface interfacename.

class classname **extends** superclass **implements** interface1, interface2,….

```
{
        body of classname
}
```


VIII. Resources required (Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer i3 RAM minimum 2 GB	As per batch size	
2	Software	jdk1.8.0 or above.		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Demonstrate the use of interfaces to implement the concept of multiple inheritance.

Note: Attach the code at the end.

```
interface Printable{

void print();

}

interface Showable{

void show();

}

class A7 implements Printable,Showable{

public void print(){System.out.println("Hello");}

public void show(){System.out.println("Welcome");}


public static void main(String args[]){

A7 obj = new A7();

obj.print();

obj.show();

 }

}
```

XI. Result (Output of Code):

Hello
Welcome

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Differentiate between Class and Interface.
2. Write similarities between interfaces and classes.
3. Write advantages of interfaces.
4. Will this code compile successfully?(Assume Suitable Class)

```
public interface FamousLine
{
    void ShowLine( )
    {
    System.out.println("Show Line");
    }
}
```

(Space for answer)

1. Class	Interface
Class can instantaite variable & Create object.	Interface can not instantiate variables & Create an object
Class can contain concrete methods	The interface can not contain concrete methods
The access specifiers used with classes are private, protected and public	The interface only one public specificer is used.

2. The similarities are:
   a. Both are java basic object types.
   b. Both can contain variables and methods.
   c. Both be inherited using Inheritance.

3. Advantages of interfaces :

   1) through interfaces we can implement multiple inheritance in java.
   2) Interfaces function to break up the complex designs and clear the dependencies between objects.
   3) Interfaces makes your application loosely coupled.

XIII. Exercise:

1 Correct the code to rectify the compile time error thrown.

```
interface NewShape
{
    void draw( );
}
interface Circle extends NewShape
{
    void getRadius( );
    int radius=10;
}
class NewCircle implements Circle
{
    public void getRadius( )
    {
            System.out.println(radius);
    }
}
class ExtendInterface extends NewCircle
{
    public static void main(String aregs[])
    {
            Circle nc=new NewCircle( );
            nc.getRadius( );
    }
}
```


2 Develop a program to find area of rectangle and circle using interfaces.

3 Write output of the program.

```
interface Pet
{
    public void test();
}
class Dog implements Pet
{
     public void test()
    {
        System.out.println("Interface Method Implemented");
    }
    public static void main(String args[]){
    Pet p = new Dog();
    p.test();
}
```

(Space for answer)

.....

.....

.....

2. interface area

```
{
    double pi = 3.14;
    double calc(double x,double y);
}

class rect implements area
{
  public double calc(double x,double y)
  {
     return(x*y);
  }  }

class cir implements area
{
  public double calc(double x,double y)
  {
     return(pi*x*x);
  }  }
class test7
{
  public static void main(String arg[])
  {
     rect r = new rect();
     cir c = new cir();
     area a;
     a = r;
     System.out.println("\nArea of Rectangle is :" +a.calc(10,20));
     a = c;
     System.out.println("\nArea of Circle is :" +a.calc(15,15));
  }  }
```

Practical No. 20: Develop a program to import different classes in package.

I. Practical Significance:

Packages are collection of classes and interfaces. Importing of Packages which helps in Reusability. Better organization of the classes and interfaces helps in resolving the name conflicts.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java."

The practical is expected to develop the following skills:

1. Write a program to create user defined packages.
2. Write a program to import the user defined packages.

IV. Relevant Course Outcome(s)- Develop program using OO method in java.

Apply concept of inheritance for code reusability.

V. Practical Outcome (PrOs)

Develop a program to import different classes in package.

VI. Relevant Affective domain related Outcome(s)

1. Follow Safety Practices
2. Practice good housekeeping
3. Demonstrate Working as a leader/a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

What is Package

It is a collection of similar types of **classes, interfaces** and **sub-packages**.

Packages are categorized in two form, **built-in package** and **user-defined package**

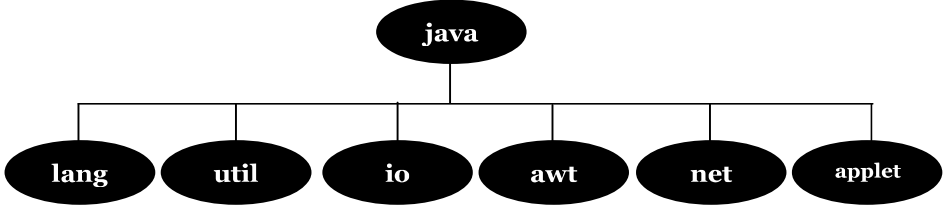


Fig. 7 Frequently Used API Packages

Using System Packages

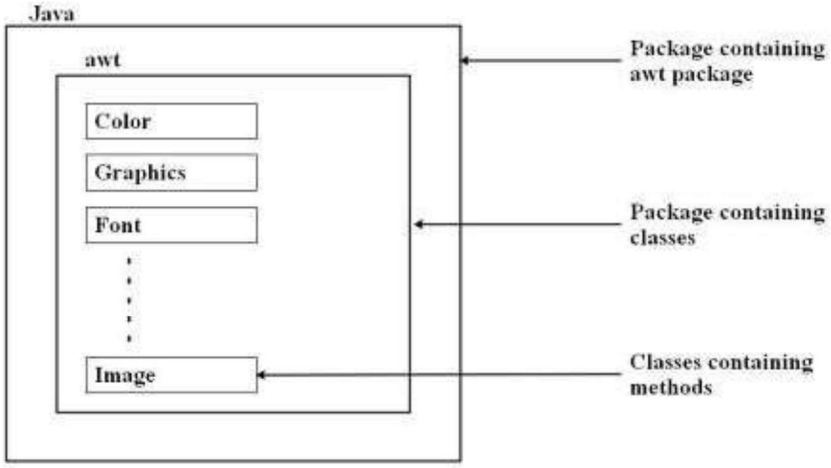


Fig. 8 Hierarchical representation of java.awt package

Accessing the classes stored in a package:

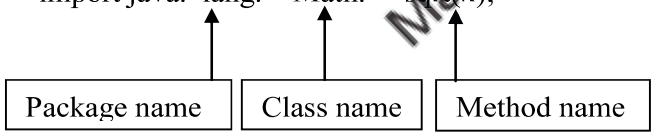
Method 1: Using import Statements:

1. The first method is to use the **Fully qualified class name** of the class:
   import pacakgename.Classname;

Example:

import java.lang.Math;

This statement imports the class Math and therefore class name can be used directly. It is not necessary to use the package name to qualify the class.



Method 2:Shortcut Approach

import packagename.*;

import java.lang.*;

This statement imports **every class** contained in the specified package.

Above statement will bring **all the classes** of java.lang package.

Creating Packages

```
package PackageName;          //package declaration
public class FirstClass        // class definition
{
        ------------------// (body of class)
}
```

Package Hierarchy

Package firstPackage.secondPackage;

Accessing a Package
A java system package can be accessed either using a **fully qualified class name** or using a **shortcut approach** through the import statement.

Syntax:
import package1 [. package2][.package3].classname;
package1 is the name of the **outer package**, package2 is the name of the package that is **inside the package1**.
Package hierarchy consists of any number of packages. Finally the explicit classname is specified.

Example:
import firstPackage.secondPackage.MyClass; // fully qualified class name
OR
import packagename.*;
import firstPackage.*; //shortcut approach

VIII. Resources required(Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer i3 RAM minimum 2 GB	As per batch size	
2	Software	jdk1.8.0 or above.		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Write a program to implement user defined packages in terms of creating a new package and importing the same.
Note: Attach the code at the end.

```
package MyPackage;
 public class Compare {
  int num1, num2;
  Compare(int n, int m) {
     num1 = n;
     num2 = m;
 }
 public void getmax(){
    if ( num1 > num2 ) {
       System.out.println("Maximum value of two numbers is " + num1);
  }
```

```
        else {
            System.out.println("Maximum value of two numbers is " + num2);
          }
        }
        public static void main(String args[]) {
            Compare current[] = new Compare[3];
            current[1] = new Compare(5, 10);
            current[2] = new Compare(123, 120);
             for(int i=1; i < 3 ; i++)
                {
                current[i].getmax();
                } }}
```

XI. Result (Output of Code):

Maximum value of two numbers is 10
Maximum value of two numbers is 123

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Name some of java packages.
2. Can we import same class/package twice? Will the JVM load the package twice at run time?
3. Write fully qualified and shortcut class naming approach with examples.

(Space for answer)

1. Some Name of Java Packages.
   1. java.text
   2. java.security
   3. java.nio
   4. java.lang
   5. java.math
   6. java.rmi
   7. java.net
   8. java.net

2. Yes, We can import a class twice in Java, it doesn't create any issues but, irrespective of the number of times you import, JVM loads the class only once.

XIII. Exercise (Any two)

1. The code uses the class defined below. Class Importclass is not defined in circle folder. Will the code run without giving any errors?

```
import circle.NewCircle;
class ImportClass
{
   public static void main(String args[ ])
   {
            circle.NewCircle nc=new circle.NewCircle( );
            System.out.println("Hello Java");
   }
}
```

2. Define a package named myInstitute include class named as department with one method to display the staff of that department. Develop a program to import this package in a java application and call the method defined in the package.
3. Develop a program which consists of the package named let_me_calculate with a class named calculator and a method named add to add two integer numbers. Import let_me_calculate package in another program (class named Demo) to add two numbers.

(Space for answer)

3.
```
public class Calculator
{
 public void add()
 {
  int a=20;
  int b=30;
  int c;
  c=a+b;
  System.out.println("Addition = "+c);
 }
}
import let_me_calculate.*;
class Demo
{
 public static void main(String args[])
 {
  Calculator obj=new Calculator();
  obj.add();
 }
}
```

2.
```
package myInstitute;
public class Department
{
  public void display()
  {
   int sr=1;
   String Name="Ajay";
   long no=987654321;
   String desig="H.O.D";
   System.out.println("Details of Staff are");
   System.out.println("Srno = "+sr);
   System.out.println("Name = "+Name);
   System.out.println("Designation = "+desig);
   System.out.println("Phone Number = "+no);
  }
}
import myInstitute.*;
class JavaApplication
{
  public static void main(String args[])
  {
    Department obj = new Department();
    obj.display();
  }
}
```

Practical No. 21 & 22: Develop a program for implementation of multithreading operation Part-I and Part-II.

I. Practical Significance:

Multithreading technique in java helps to run multiple programs or a processes concurrently by utilizing the maximum CPU time. Multithreaded technique is implemented by creating, declaring, extending, implementing by thread. Student will be able to implement different types of thread methods by assigning the priority to illustrate simultaneous execution of thread operation.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java."

The practical is expected to develop the following skills:

1. Develop program that enables programmers to do multiple things at one time using multiple threads.
2. Write programs to extend the thread and implement various thread methods.

IV. Relevant Course Outcome(s)

Develop programs using multithreading.

V. Practical Outcome (PrOs)

Develop a program for implementation of multithreading operation.

VI. Relevant Affective domain related Outcome(s)

1. Follow Safety Practices
2. Practice good housekeeping
3. Demonstrate Working as a leader/a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

Multithreading: Multithreading is a small program (process) which is divided into two or more subprogram (processes), that can be implemented simultaneously.

Thread: It is a small process which is used to divide a program into number of sub parts and each part can be executed in parallel.

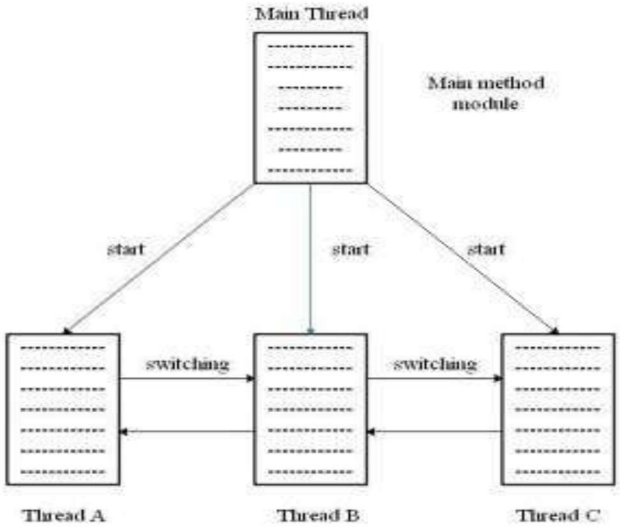


Fig. 10 A Multithreaded Program

Creating Threads

Threads are implemented using a method called run( ).

Syntax:

```
public void run ( )
{
        ……..
        ……..
}
```

Statements for implementing thread

A thread can be created in two ways:

1. **By creating a thread class:**
   Define a class that extends Thread class and override its run( )
   Extending the thread Class

```
class MyThread extends Thread
{
……..
……..
}
```

2. **By converting a class to a thread:**
   Define a class that implements Runnable interface. The Runnable interface has only one method, run( )

```
class A implements Runnable
{
    ……..
    ……..
}
```

   Synchronization
   It means only a single thread can execute a block of code at the same time.

Example:
The method that will read information from a file and the method that will update the same file may be declared as synchronized.

```
synchronized void update( )
{
    ……..
    ……..// synchronized code
}
```

Java Thread Method:

Sr. No.	Type	Method	Description
1	void	start( )	It is used to start the execution of the thread.
2	void	run( )	It is used to perform action for a thread.
3	static void	sleep( )	It sleeps a thread for specified amount of time.
4	static Thread	currentThread( )	It returns a reference to the currently executing thread object.
5	void	join( )	It waits for a thread to die.
6	int	getPriority( )	It returns the priority of the thread.
7	void	setPriority( )	It changes the priority of the thread.
8	string	getName( )	It returns the name of the thread.
9	void	setName( )	It changes the name of the thread.
10	long	getID( )	It returns the id of the thread.
11	boolean	isAlive( )	It tests if the thread is alive.
12	static void	yield( )	It causes the currently executing thread object to temporarily pause, allow other threads to execute.
13	void	suspend( )	It is used to suspend the thread.
14	void	resume( )	It is used to resume the suspended thread.
15	void	stop( )	It is used to stop the thread.
16	void	destroy( )	It is used to destroy thread group , all of its subgrp.
17	void	notify( )	It is used to give the notification for only one thread which is waiting for a particular object.
18	void	notifyAll( )	It is used to give the notification to all waiting threads of a particular object.

VIII. Resources required (Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer i3 RAM minimum 2 GB	As per batch size	
2	Software	jdk1.8.0 or above.		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Develop a simple real-life application program to illustrate the use of multithreads.

Note: Attach the code at the end.

```
public void run()
{
 try {
System.out.println(
   "Thread " + Thread.currentThread().getId()
   + " is running");
 }
 catch (Exception e) {
  System.out.println("Exception is caught");
 }
}
}
public class Multithread {
 public static void main(String[] args)
 {
  int n = 1;
  for (int i = 0; i < n; i++) {
   MultithreadingDemo object
    = new MultithreadingDemo();
   object.start();
  }}}
```

XI. Result (Output of Code):

Thread 11 is running

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Is it possible to start a thread twice?
2. Can we call the run() method instead of start()?
3. Differentiate between notify( ) and notifyAll( )?
4. Explain the use of keyword synchronized.

(Space for answer)

1. No. After starting a thread, it can never be started again. If you does so, an IllegalThreadStateException is thrown. In such case, thread will run once but for second time, it will throw exception.

2. Each thread starts in a separate call stack.
Invoking the run() method from main thread, the run() method goes onto the current call stack rather than at the beginning of a new call stack.

3. notify	notifyAll
1. In case of multiThreading notify() method sends the notification to only one thread among the multiple waiting threads which are waiting for lock.	1. While notifyAll() methods in the same context sends the notification to all waiting threads instead of single one thread.
2. As in case of notify the notification is sent to single thread among the multiple waiting threads so it is sure that which of those waiting thread is going to receive the lock.	2. On other hand notifyAll sends notification to all waiting threads hence it is not clear which of the thread is going to receive the lock.

4. Use of keyboard synchronized :

Purpose into only allow one thread at a time into a particular section of code, Thus allowing use to protect.

XIII. Exercise:

1. Implement multithreading to perform simultaneous processes.
2. The code give below calls the run( ) method of two threads while setting their priority. Will this code compile successfully? If not, correct the code.

```
class t1 extends Thread
{
   public void run( )
   {
            System.out.println("This is Thread1 class");
   }
}
Class t2 extends Thread
{
   public void run( )
   {
            System.out.println("This is Thread2 class");
   }
}

public class ThreadP
{
   public static void main(String args[ ])
   {
            t1 t=new t1( );
            t2 tt=new t2( );
            t.setPriority(Thread.MIN_PRIORITY);
            tt.setPriority(Thread.MIN_PRIORITY);
            t1.run( );
            t2.run( );}
   }
```


3. Create three threads and run these threads according to set priority.

(Space for answer)

1.
```
class Thread1 extends Thread
{  public void run() {
   System.out.println("Thread1 is  running "); }}
class Thread2 extends Thread{
   public void run()  {
    System.out.println("Thread2 is  running ");  }}
public class Threads{
    public static void main(String args[]){
     Thread1 obj1=new Thread1();
     Thread2 obj2=new Thread2();
     obj1.run();
     obj2.run();
    }}
```

3.
```
class t1 extends Thread
 {
  public void run()
  {
   System.out.println("This is Thread1 class");
  }
 }
 class t2 extends Thread
 {
  public void run()
  {
   System.out.println("This is Thread2 class");
  }
 }
 class t3 extends Thread
 {
  public void run()
  {
   System.out.println("This is Thread3 class");
  }
 }
 public class Threads
 {
  public static void main(String args[])
  {
   t1 t=new t1();
   t2 tt=new t2();
   t3 ttt=new t3();
   t.setPriority(Thread.MIN_PRIORITY);
   tt.setPriority(Thread.MAX_PRIORITY);
   ttt.setPriority(Thread.NORM_PRIORITY);
   t.start();
   tt.start();
   ttt.start();
  }
 }
```

Practical No. 23, 24 & 25: Develop a program for implementation of try, catch and finally block.

I. Practical Significance:

Managing errors and Exception handling helps to detect exceptional conditions in a program and fix the exceptions as and when they occur. Student will be able to handle different types of exceptions using try, catch and finally blocks.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java."

The practical is expected to develop the following skills:

1. Develop a program to generate and process the exception using try and catch block.
2. Write a program to execute the finally block.

IV. Relevant Course Outcome(s)

Implement Exception Handling.

V. Practical Outcome (PrOs)

Develop a program for implementation of try, catch and finally block.

VI. Relevant Affective domain related Outcome(s)

1. Follow Safety Practices
2. Practice good housekeeping
3. Demonstrate Working as a leader/a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

Errors:

Errors are the mistakes that can make a program go wrong.

Error may produce a wrong results or abruptly terminates the execution of the program or may cause the system to crash.

Detecting and managing the errors is very important during the program execution.

Types of Errors:

1. Compile-time errors
2. Run-time errors

Exception Handling Tasks:

1. Find the Problem (Hit the exception)
2. Inform that an error has occurred (Throw the exception)
3. Receive the error information (Catch the exception)
4. Take corrective actions (Handle the exception)

Common Java Exceptions

Sr. No.	Exception Type	Cause of Exception
1	ArithmeticException	Caused by math errors such as division by zero
2	ArrayIndexOutOfBoundsException	Caused by bad array indexes
3	ArrayStoreException	Caused when a program tries to store the wrong type of data in an array
4	FileNotFoundException	Caused by an attempt to access a nonexistent file
5	IOException	Caused by general I/O failures, such as inability to read from a file
6	NullPointerException	Caused by referencing a null object
7	NumberFormatException	Caused when a conversion between strings and number fails
8	OutOfMemoryException	Caused when there's not enough memory to allocate a new object
9	SecurityException	Caused when an applet tries to perform an action not allowed by the browser's security setting
10	StackOverflowException	Caused when the system runs out of stack space
11	StringIndexOutOfBoundsException	Caused when a program attempts to access a nonexistent character position in a string

Categories of Exceptions:

1. **Checked Exceptions**
   Checked exceptions are explicitly handled in the code itself using try catch blocks.
   These are extended from the java.lang.Exception class
2. **Unchecked Exceptions**
   Unchecked exceptions are not necessarily handled in the program code, instead the JVM handles such exceptions.
   These are extended from the java.lang.Runtime.Exception class.

Exception Handling Code:

Try:
A keyword 'try' is used for a block of code that causes an error condition and 'throw' an exception
Catch:
A keyword 'catch' is used for a block of code that 'catches'the exception thrown by the 'try' block and handles it properly

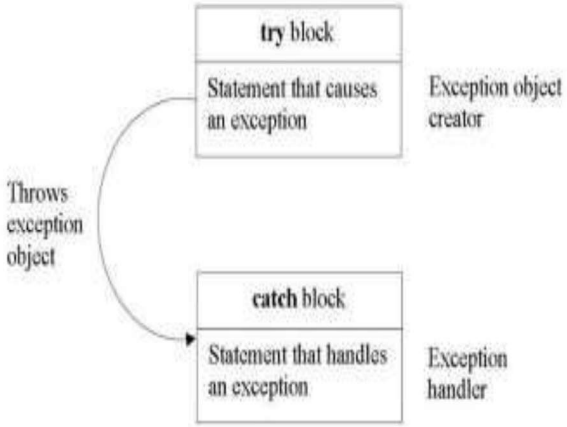


Fig. 12 Exception Handling Mechanism

Syntax: Simple try, catch and finally statement

```
try
{
        Statement;              //generates an exception
}
catch(Exception-type  e)
{
        Statement;              //processes the exception
}
……….
finally {……..} //optional
```


Finally Statement:-

finally statement handles an exception that is not caught by any of the earlier catch statements.

It is used to handle any exception generated within a try block.

It is written immediately after the try block or after the last catch block.

VIII. Resources required(Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer i3 RAM minimum 2 GB	As per batch size	
2	Software	jdk1.8.0 or above.		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Demonstrates exception handling using try, catch and finally block.

Note: Attach the code at the end.

(Space for answer)

```
class Main
{
  public static void main(String[] args)
{
    try {
      int divideByZero = 5 / 0;
    }
catch (ArithmeticException e){
    System.out.println("ArithmeticException => " +
e.getMessage());
    }
      finally {
      System.out.println("This is the finally block");
    }
  }
}
```


XI. Result (Output of Code):

```
ArithmeticException => / by zero
This is the finally block
```

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. How exception is thrown by main method?
2. Differentiate between error and exception in java.
3. Can we throw exception manually? Illustrate with sample program.
4. Explain the use of finally block?

(Space for answer)

1.When exception is thrown by main() method, Java Runtime terminates the program and print the exception message and stack trace in system console. The throws clause only states that the method throws a checked FileNotFoundException and the calling method should catch or rethrow it.

2. error	exception
Classified as an unchecked type	Classified as checked and unchecked
It belongs to java.lang.error	It belongs to java.lang.Exception
It is irrecoverable	It is recoverable
It can't be occur at compile time	It can occur at run time compile time both
Ex : OutOfMemoryError IOError	Ex. NullPointerException SqlException

4. The finally block in java is used to put important codes such as clean up code e.g. closing the file or closing the connection.
The finally block executes whether exception rise or not and whether exception handled or not.
A finally contains all the crucial statements regardless of the exception occurs or not.

XIII. Exercise:

1. The program calculates sum of two numbers inputted as command-line arguments. When will it give an exception?

```
class excep
{
   public static void main(Sting args[ ])
   {
        try
        {
             int n=Integer.parseInt(args[0]);
             int n1=Integer.parseInt(args[1]);
             int n2=n+n1;
             System.out.println("Sum is:"+n2);
        }
        catch(NumberFormatException ex)
        {
             System.out.println(ex);
        }
        finally
        {
             System.out.println("You inputted a correct integer number");
        }
   }
}
```


2. Develop a program to accept a password from the user and throw "Authentication Failure" exception if the password is incorrect.
3. Write the exception thrown by the following code block?

```
Integer[][] ints = { { 1, 2, 3 }, { null }, { 7, 8, 9 } };
    System.out.println("value = " + ints[1][1].intValue());
```

(Space for answer)

```
2. class AuthenticationException extends Exception {
   public AuthenticationException(String message) {
    super(message);
   }
   }
   public class AuthenticationExcDemo {
   public static void main(String[] args) {
    InputStreamReader isr = new
  InputStreamReader(System.in);
    BufferedReader br = new BufferedReader(isr);
    String pwd;
     try {
       System.out.print("Enter password :: ");
     pwd = br.readLine();
      if(!pwd.equals("123"))
      throw new AuthenticationException("Incorrect
 password\nType correct password");
      else
       System.out.println("Welcome User !!!");
       }
     catch (IOException e) {
      e.printStackTrace();
     }
     catch (AuthenticationException a) {
      a.printStackTrace();
     }
     System.out.println("BYE BYE");
    }
    }
```

Practical No. 26 & 27: Develop a program for implementation of throw and throws clause.

I. Practical Significance:

The throw and throws clause is used to explicitly throw an exception from a method or any block of code. It is mainly used for throwing the custom exceptions (User defined exceptions). Students will be able to throw the user defined exceptions.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java."

The practical is expected to develop the following skills:

1. Develop a program to throw our own exceptions i.e. user defined exceptions using throw.
2. Write a program to predict explicitly certain kinds of exception using throws clause.

IV. Relevant Course Outcome(s)

Implement Exception Handling.

V. Practical Outcome (PrOs)

Develop a program for implementation of throw & throws clause.

.

VI. Relevant Affective domain related Outcome(s)

1. Follow Safety Practices
2. Practice good housekeeping
3. Demonstrate Working as a leader/a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

Throwing Our Own Exceptions

throw: All methods in java use the 'throw' statement explicitly to **throw** an exception from a method or any block of code The **throw** is a keyword in **java**. **throw** can be used for either checked or unchecked exception. It is mainly used to **throw** custom exceptions.

Syntax:

throw Throwable-instance;

where Throwable-instance must be an object of type Throwable or a subclass of Throwable.
throws: It is a keyword in java. Its is used to declare an exception. It is used with a method signature. Multiple exceptions can be declared through throws.

Syntax:
type method name(parameters) throws exception list

VIII. Resources required (Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer i3 RAM minimum 2 GB	As per batch size	
2	Software	jdk1.8.0 or above.		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Demonstrate use of throw and throws clause.
Note: Attach the code at the end.


```
class ThrowExcep
{ static void fun()
  { try
    { throw new NullPointerException("demo");}
    catch(NullPointerException e){
      System.out.println("Caught inside fun().");
      throw e;
    } }
  public static void main(String args[]){
    try{
      fun();}
    catch(NullPointerException e) {
      System.out.println("Caught in main.");
    }}}
```

XI. Result (Output of Code):

Caught inside fun().
Caught in main.

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Differentiate between throw and throws clause.
2. In which situation the throws clause is used?
3. Write the simple program for throwing our own exceptions.

(Space for answer)

1.

Throw	Throws
1.Java throw keyword is used to explicitly throw an exception.	1. Java throws keyword is used to declare an exception.
2. Checked exception cannot be propagated using throw only.	2. Checked exception can be propagated with throws
3. Throw is followed by an instance.	3. Throws is followed by class.
4. Throw is used within the method.	4. Throws is used with the method signature.

2. The Java throws keyword is used to declare an exception. It gives an information to the programmer that there may occur an exception so it is better for the programmer to provide the exception handling code so that normal flow can be maintained. Exception Handling is mainly used to handle the checked exceptions.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

XIII. Exercise:

1. Define an Exception called "NotMatchException" that is thrown when a string is not equal to "India". Write a program that uses this exception.
2. Write the output of the following program:

```
class tst
{
   public static void main(String args[])
   {
      Thread.sleep(10000);
      System.out.println("Hello java");
   }
}
```

3. Write the output of the following program:

```
class tst
{
   public static void main(String[] args)throws InterruptedException
   {
      Thread.sleep(10000);
      System.out.println("Hello Java");
   }
}
```

(Space for answer)

1.
```
class StringMismatchException extends Exception {
 public StringMismatchException(String str) {
  System.out.println(str);
 }
}
public class StringExc {
 public static void main(String[] args) {
Scanner scan = new Scanner(System.in);
 System.out.print("Enter the string :: ");
 String input = scan.nextLine();
  try {
 if(input.equalsIgnoreCase("India"))
  System.out.println("String matched !!!");
 else
  throw new StringMismatchException("String not
matched ???");
  }
  catch (StringMismatchException s) {
   System.out.println(s);
  }
 }
 }
```

2. error: unreported exception InterruptedException; must be caught or declared to be thrown

3. Hello Java

Practical No. 28: Develop minimum two basic Applets. Display output with applet viewer and browser.

a) **Develop a program on basic applet.**
b) **Develop a program using control loops in applets**

I. Practical Significance:

Java applets help to make the web pages more dynamic. Java is also used to create small, dynamic programs that run along with or are embedded within Web pages. These programs (applets) can be used to display maps, weather, games or other interactive widgets or tools on a Web page.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline Knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and Practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java."

The practical is expected to develop the following skills:

1. Write a program to build applet code.
2. Develop a program using different states of an applet and execute using applet viewer.

IV. Relevant Course Outcome(s)

Develop program using graphics and Applet.

V. Practical Outcome (PrOs)

Develop minimum two basic Applets. Display output with applet viewer and browser.

1. Develop a program on basic applet.
2. Develop a program using control loops in applets.

VI. Relevant Affective domain related Outcome(s)

1. Follow Safety Practices
2. Practice good housekeeping
3. Demonstrate Working as a leader/a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

Applets are small Java programs that can be used in Internet Computing. Applets can be transported over the Internet from one computer to another and run using the **Applet Viewer** or any Web browser that supports Java.

Through Applets Arithmetic operations, display graphics, play sounds, accept user input, animation and play interactive games can be performed.

java.awt.Component class

public void paint(Graphics g): It is used to paint the Applet. It contains Graphics class object which is used for drawing oval, rectangle, arc, etc.

Simple Program: The HelloJava applet

```
import java. awt. *;
import java. applet.*;
public class HelloJava extends Applet
{
public void paint(Graphics g)
{
        g.drawString("Hello Java",10,100);
}
}
```

Output

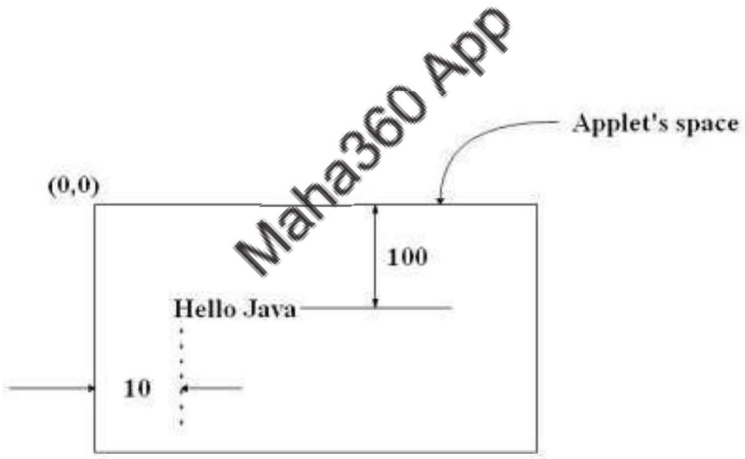


Fig. 14 Output of Program

Applet Tag

```
<APPLET
        CODE=HelloJava.class
        WIDTH=400
        HEIGHT=200>
</APPLET>
```

Adding Applet to HTML file

```
<HTML>
<! This page includes a welcome title in the title bar and also displays a welcome
message. Then it specifies the applet to be loaded and executed>
```

```
<HEAD>
        <TITLE>
                Welcome to Java Applets
        </TITLE>
</HEAD>
<BODY>
        <CENTER>
                <H1>Welcome to the World of Applets</H1>
        </CENTER>
        <BR>
        <CENTER>
                <APPLET
                        CODE=HelloJava.class
                        WIDTH=400
                        HEIGHT=200>
                </APPLET>
        </CENTER>
        </BODY>
</HTML>
```

Note: Save this file as **HelloJava.html** and save it in the same directory as the compiled applet.

Creating an Executable Applet:
Executable applet is obtained by compiling the source code of the applet i.e. **.class** file of the applet.
Compiling an applet is same as compiling an application. Java compiler is used to compile the applet.
HelloJava applet is stored in a file called **HelloJava.java**
Compile the program using **javac HelloJava.java**
The compiled output file is created called **HelloJava.class** (if no errors).

Running the Applet:
In above sample codes we have created **applet files** as well as the **HTML file containing the applet,** we must have the following files in our current directory.

HelloJava.java
HelloJava.class
HelloJava.html

There are two ways to run an applet:

1. By Java-enabled web browser tool(i.e. HTML file)
2. By Java appletviewer tool

1. By Java-enabled web browser tool (i.e. HTML file)

Through Java-enabled Web browser the entire web page can be seen containing the applet.

```
c:\>javac HelloJava.java
c:\>appletviewer HelloJava.java
```

2. By Java appletviewer tool

Through appletviewer tool, only applet output can be seen.
The appletviewer is not a full-fledged web browser and therefore it ignores all of the HTML tags except the part pertaining to the running of the applet.

To run the applet:
appletviewer HelloJava.html

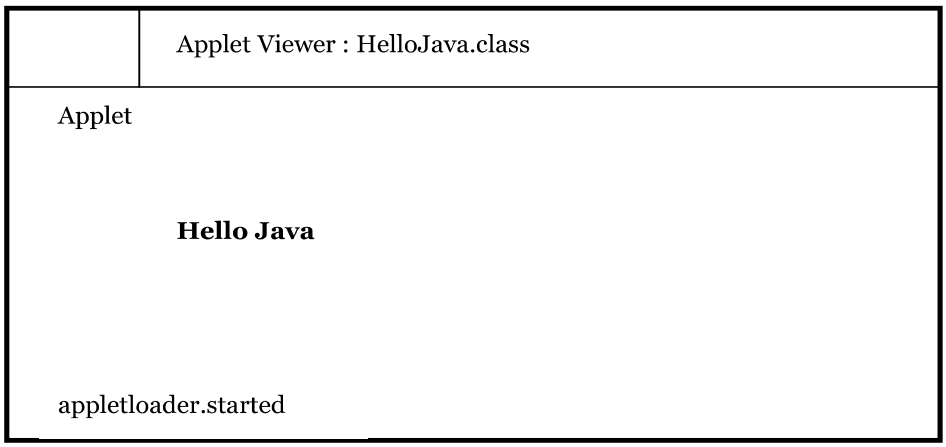


Fig.15. Output of HelloJava applet by using appletviewer

VIII. Resources required(Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer i3 RAM minimum 2 GB	As per batch size	
2	Software	jdk1.8.0 or above.		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Create two applets by using different ways in java.

```
//First.java
import java.applet.Applet;
import java.awt.Graphics;
public class First extends Applet{
public void paint(Graphics g){
g.drawString("welcome",150,150);
}
}
```


XI. Result (Output of Code):

welcome

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Differentiate between init( ) and start( )methods.
2. Explain use of start( ) and stop( ) method in applet life cycle?
3. The method paint( ) belongs to which class?

(Space for answer)

1. init( )	start( )
1. init( ) is used to initialized the Applet.	1. start( ) is invoked after the init() method or browser is maximized.
2. It is invoked only once	2. It is used to start the Applet.

2. A start() method is often used to start any threads the applet will need while it runs.
   The stop() method is called at least once in an applet's life, when the browser leaves the page in which the applet is embedded

3. The Method Paint () belongs to type Graphics class.

XIII. Exercise:

1. Develop a basic applet to display "Welcome to the World of Applet".
2. Develop a program to implement all methods of applet.
3. Develop a program using control loops in applets.

(Space for answer)

1. package Applet;

```
import java.applet.*;
import java.awt.*;

public class WelcomeToApplet extends Applet
{
    public void paint (Graphics g)
    {
        g.drawString ("Welcome to the World of Applet ", 25, 50);

    }
}
```

3.
```
import java.awt.*;
import java.applet.*;

public class ControlLoopApplet extends Applet
{
 public void paint(Graphics g)
 {
  for(int i=1;i<=4;i++)
 {
 if(i%2==0)
 {
  g.fillOval(90,i*50+10,50,50);
  g.setColor(Color.black);
 }
 else
 {
  g.drawOval(90,i*50+10,50,50);
  g.setColor(Color.red);
 }
 }
 }
}
```

Practical No. 29: Write a program to create animated shape using graphics and applets. You may use following shapes:

a) Lines and Rectangles
b) Circles and Ellipses.
c) Arcs
d) Polygons with fill Polygon method.

I. Practical Significance:

The Graphics class of java provides methods for drawing many different types of shapes. Students will be able to use different methods of graphics programming to draw simple lines, figures of different shapes, images and text in different fonts and styles with different appearances of colors.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java."

The practical is expected to develop the following skills:

1. Write programs by using different methods of graphics class.
2. Develop applet programs using graphics programming

IV. Relevant Course Outcome(s)

Develop program using graphics and Applet.

V. Practical Outcome (PrOs)

Write a program to create animated shape using graphics and applets. You may use following shapes:

1. Lines and Rectangles
2. Circles and Ellipses.
3. Arcs
4. Polygons with fill Polygon method.

VI. Relevant Affective domain related Outcome(s)

1. Follow Safety Practices
2. Practice good housekeeping
3. Demonstrate Working as a leader/a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

Every applet has its own area of the screen known as canvas, where it creates its display.

The size of an applet's space is decided by the attributes of the <APPLET… >tag.

Drawing Methods of the Graphics Class:

Sr. No.	Method	Description	Syntax
1	drawArc()	Draws a hollow arc	void drawArc(int top, int left, int width, int height, int startAngle, int
2	drawLine()	Draws a straight line	void drawLine(int startX, int startY, int endX, int endY)
3	drawOval()	Draws a hollow oval	void drawOval(int top, int left, int width, int height)
4	drawPolygon()	polygon	void drawPolygon(int ,[] , int y[], int numPointer)
5	drawRect()	Draws a rectangle	void drawRect(int top, int left, int width, int height)
6	drawRoundRect()	Draws a hollow rectangle with rounded corners	void drawRoundRect(int top, int left, int width, int height int Xdiam, int YDiam)
7	drawString()	Displays a text string	void drawString(String str, int x, int y)
8	fillArc()	Draws a filled arc	void fillArc(int top, int left, int width, int height, int startAngle, int sweetAngle)
9	fillOval()	Draws a filled oval	void fillOval(int top, int left, int width, int height)
10	fillPolygon()	Draws a filled polygon	void fillPolygon(int ,[] , int y[], int numPointer)
11	fillRect()	Draws a filled rectangle	void fillRect(int top, int left, int width, int height)
12	fillRoundRect()	Draws a filled rectangle with rounded corners	void fillRoundRect(int top, int left, int width, int height int Xdiam, int YDiam)
13	getColor()	Retrieves the current drawing color	void getColor(String nm)
14	getFont()	Retrieves the currently used font	void getFont( )
15	getFontMetrics()	Retrieves the information about the	void getFontMetrics( )
16	setColor()	Sets drawing color	void setColor(Color c)
17	setFont()	Sets the font	void setFont( )

VIII. Resources required (Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer i3 RAM minimum 2 GB	As per batch size	
2	Software	jdk1.8.0 or above.		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

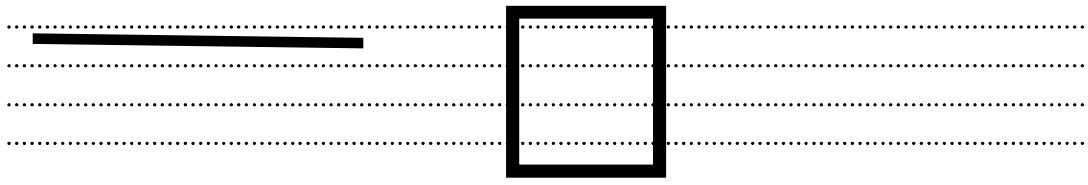
Write a program to implement an applet to draw basic animated shapes.

Note: Attach the code at the end.

```
import java.awt.*; // Importing awt package

import java.applet.*; // Importing applet package
public class Shapes extends Applet
{
public void paint(Graphics g)
{
g.setFont(new Font("Cambria", Font.BOLD,15));
g.drawString("Different Shapes", 15, 15);
g.drawLine(10,20,50,60);
g.drawRect(10,70,40,40);
}
}
```


XI. Result (Output of Code):



XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Differentiate between executing the applet with appletviewer and HTML file.
2. Explain methods required to draw different shapes with different colors?
3. Differentiate between setforeground( ) and setColor( ) methods.
4. Differentiate between applets and applications.

(Space for answer)

4. applets	applications
1. Applets are small Java programs that are designed to be included with the HTML web document. They require a Java-enabled web browser for execution.	1. Applications are just like a Java programs that can be execute independently without using the web browser.
2. Applet does not require a main function for its execution.	2. Application program requires a main function for its execution.
3. Applets don't have local disk and network access.	3. Application programs have the full access to the local file system and network.

3. setforeground()	setcolour ( )
1. setforeground() sets the color of the text	1. setcolor() sets the color of the drawing
2. Rather setforeground() corresponds to color of the text and to be clear it is no way related to the background color.	2. "setcolor()" will set the pen color of the drawing which is going to be drawn
3. Ex : setForeground(Color.red); drawstring("Hello", 100, 80); The text "Hello" will appear in red.	3. Ex : setColor(Color.BLUE); drawOval(20,20,100,100); fillOval(20, 20, 100, 100);

1. When you see a HTML page with an applet in a Java-enabled web browser, the applet code gets transferred to the system and is finally run by the Java-enabled virtual machine on the browser. Applets are also compiled using the javac command but can only run using the appletviewer command or with a browser.

XIII. Exercise (Any two)

.

1. Develop a program to draw a polygon.
2. Develop an applet for drawing a human face.
3. Write the output of the program.

```
import java.awt.*;
import java.applet.*;
/*
<applet code="Arcs" width=300 Height=300>
</applet>
*/
public class Arcs extends Applet
{
    public void paint(Graphics g)
    {
        g.drawArc(10,40,70,70,0,75);
        g.fillArc(100,40,70,70,0,75);
        g.drawArc(10,100,70,80,0,175);
        g.fillArc(100,100,70,90,0,270);
        g.drawArc(200,80,80,80,0,180);
    }
}
```


(Space for answer)

```
1. import java.awt.*;
   import java.applet.*;
   public class DrawingPolygons extends Applet
   {
   public void paint(Graphics g)
   {
   int x[] = { 70, 150, 190, 80, 100 };
   int y[] = { 80, 110, 160, 190, 100 };
   g.drawPolygon (x, y, 5);
   }
   }
```


XIV. References/ Suggestions for Further Reading

1. https://www.javatpoint.com/
2. https://javaproglang.blogspot.com/2014/03/java-graphics-programming.html#.W0sRDdIzbIU
3. Programming with Java-A Primer by E Balagurusamy

2.
```
import java.applet.*;
import java.awt.*;
public class Human_Face extends Applet
{
public void init()
{
setBackground(Color.white);
}
public void paint(Graphics g)
{
Color clr=new Color(255,179,86);
g.setColor(clr);
g.drawOval(100,100,250,300);
g.fillOval(100,100,250,300);
g.setColor(Color.black);
g.drawOval(160,185,40,25);
g.fillOval(160,185,40,25);
g.drawOval(250,185,40,25);
g.fillOval(250,185,40,25);
g.drawArc(160,170,35,10,0,180);
g.drawArc(250,170,35,10,0,180);
g.drawLine(210,265,210,275);
g.drawLine(240,265,240,275);
g.drawArc(210,275,30,10,0,-180);
g.drawArc(175,300,100,50,0,-180);
}
}
```

Practical No. 30: Develop a program to draw following shapes, graphics and applets.

a) Cone
b) Cylinders
c) Cube
d) Square Inside a circle
e) Circle inside a square

I. Practical Significance:

The Graphics class of java provides methods for drawing many different types of shapes. Students will be able to use different methods of graphics programming to draw cube, cylinders, cones, figures of different shapes, images and text in different fonts and styles with different appearances of colors.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills

"Develop Applications using Java."

The practical is expected to develop the following skills:

1. Write programs by using different methods of graphics class.
2. Develop applet programs using graphics programming.

IV. Relevant Course Outcome(s)

Develop program using graphics and Applet.

V. Practical Outcome (PrOs)

Develop a program to draw following shapes, graphics and applets.

1. Cone
2. Cylinders
3. Cube
4. Square Inside a circle
5. Circle inside a square

VI. Relevant Affective domain related Outcome(s)

1. Follow Safety Practices
2. Practice good housekeeping
3. Demonstrate Working as a leader/a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

A java applet draws graphical image inside the space using the coordinate system. To draw different shapes, appropriate graphics method with proper syntax is required.

VIII. Resources required (Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Qty	Remarks (If any)
1	Software	jdk 1.8.0 or above	As per batch size	

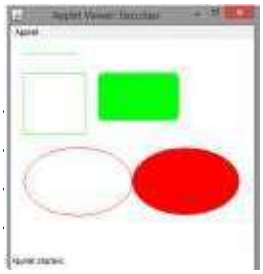
X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Demonstrate the use of basic applet, which includes different shapes like cone, cube, cylinders etc.

Note: Attach the code at the end.

```
import java.applet.*;
import java.awt.*;
public class Geo extends Applet
{
public void paint(Graphics g)
{
g.setColor(Color.GREEN);
g.drawLine(20,20,100,20);
g.drawRect(20,50,90,90);
g.fillRoundRect(130,50,120,70,15,15);
g.setColor(Color.RED);
g.drawOval(20,160,160,100);
g.fillOval(180,160,160,100);
 }
 }
/* <applet code="Geo.class" width =300
height=300>
 </applet>
 */
```

XI. Result (Output of Code):



XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Which of these methods is a part of Abstract Window Toolkit (AWT)?
   a) display( ) b) paint( ) c) drawstring( ) d) none of the above
   Ans._____
2. Enlist the methods required to draw cone/cylinder.
3. Explain the method with syntax to draw circle.
4. Differentiate Applet and application.

(Space for answer)

1. b) paint( )

2. The methods required to draw cone/cylinder.
   1. drawOval( );
   2. drawLine( );

3. The method drawOval() is one of the methods of a draw circle.
   Syntax : drawOval(int X, int Y, int width, int height)

4. applets	applications
1. Applets are small Java programs that are designed to be included with the HTML web document. They require a Java-enabled web browser for execution.	1. Applications are just like a Java programs that can be execute independently without using the web browser.
2. Applet does not require a main function for its execution.	2. Application program requires a main function for its execution.
3. Applets don't have local disk and network access.	3. Application programs have the full access to the local file system and network.

XIII. Exercise:

1. Write output of the program.

```
/*
<applet code="SetBackgroundColorExample" width=200 height=200>
</applet>
*/
import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;
public class SetBackgroundColorExample extends Applet
{
    public void paint(Graphics g){
        /*
         * Set background color of an applet using
         * void setBackground(Color c) method.
         */
        setBackground(Color.red);
    }
}
```

2. Develop a program to draw any 2 of the following shapes:
   a. Cone

b. Cylinder
c. Cube

3. Develop a program to draw any one of the following:
   a. Square Inside a circle
   b. Circle inside a square

(Space for answer)

2. a. cone :-

```
import java.applet.*;
import java.awt.*;
public class Cone extends Applet
{
public void paint(Graphics g)
{
g.drawOval(80,280,320,100);
g.drawLine(240,50,82,320);
g.drawLine(240,50,398,320);
g.drawLine(240,330,398,330);
g.drawLine(240,50,240,330);
g.drawString("Radius",260,360);
g.drawString("Height",246,255);
g.drawString("Slant Height",340,210);
g.drawString("Cone",220,420);
}}
```

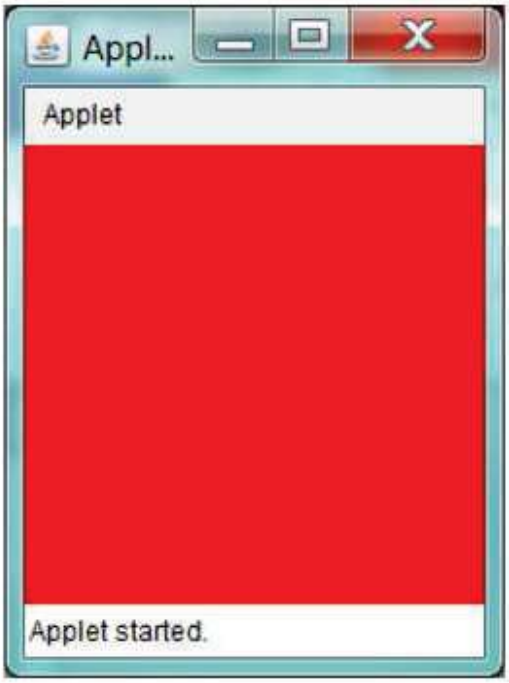
b. cylinder :-

```
import java.applet.*;
import java.awt.*;
public class Cylinder extends Applet
{
public void paint(Graphics g)
{
g.drawString("Cylinder",80,50);
g.drawOval(50,60,100,50);
g.drawLine(50,80,50,200);
g.drawLine(150,80,150,200);
g.drawOval(50,180,100,50);
}}
```

3. a. Square inside a circle

```
/*<applet code="Square.class" width=300 height=300>
</applet>
*/
import java.applet.*;
import java.awt.*;
public class Square extends Applet
{
public void paint(Graphics g)
{
g.drawString("Square inside a circle",150,115);
g.drawOval(180,10,80,80);
g.drawRect(192,22,55,55);
}
}
```


1. Output :-



Practical No. 31 & 32: Develop a program for implementation of I/O stream and file stream classes.

I. Practical Significance:

File handling is used to manage and store data which performs various operations. Reading and writing can be done at the level of bytes or characters.

Java supports byte stream classes that provide support for handling I/O operations on bytes and character stream classes for operations on characters.

II. Relevant Program Outcomes (POs)

- **Basic knowledge:** Apply knowledge of basic mathematics, sciences and basic engineering to solve the computer group related problems.
- **Discipline knowledge:** Apply Computer Programming knowledge to solve the computer group related problems.
- **Experiments and practice:** Plan to perform experiments and practices to use the results to solve the computer group related problems.
- **Engineering tools:** Apply relevant Computer programming technologies and tools with an understanding of the limitations.
- **Individual and team work:** Function effectively as a leader and team member in diverse/multidisciplinary teams.
- **Communication:** Communicate effectively in oral and written form.

III. Competency and Practical skills:

"Develop Applications using Java."

This practical is expected to develop the following skills:

1. Write the program using file class to perform different file operations.
2. Develop the program to handle Input/Output Exceptions.

IV. Relevant Course Outcome(s)

Develop programs for handling I/O and file streams.

V. Practical Outcome (PrOs)

1. Develop a program for implementation of I/O stream classes.
2. Develop a program for implementation of file stream classes.

VI. Relevant Affective domain related Outcome(s)

1. Follow Safety Practices
2. Practice good housekeeping
3. Demonstrate Working as a leader/a team member.
4. Follow ethical practices.

VII. Minimum Theoretical Background

File:

A **file** is a collection of records placed in a particular area on the disk. A **record** is composed of several fields and field is a group of characters. Storing and managing data using files is known as **file processing** which includes tasks such as **creating files, updating files** and **manipulation of data.**

Streams:
In file processing, **input** refers to the flow into a program and **output** means the flow of data out of a program.

Input to a program may come from the **keyboard**, the **mouse**, the **memory**, the **disk**, a **network** or **another program**.

Output from a program may go to the **screen**, the **printer**, the **memory**, the **disk**, a **network**, or **another program.**

A stream in java a path along which data flows (like a river or a pipe along which water flows. It has a source (of data) and a destination (for that data).

In Java, I/O operations are performed through streams.
Java streams are classified into two basic types:

1. Input Stream
2. Output Stream

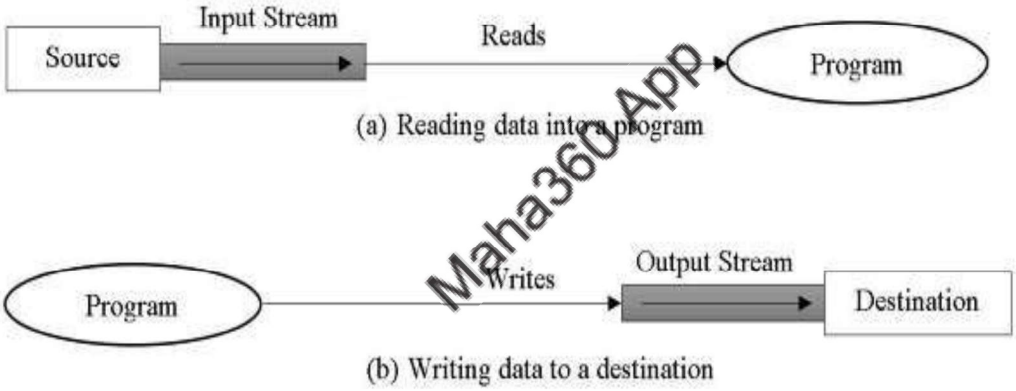


Fig. 16. Using Input and Output Streams

Categories of Stream Classes:
1. Byte stream classes
2. Character stream classes

VIII. Resources required

Nil

IX. Resources used

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer i3 RAM minimum 2 GB	As per batch size	
2	Software	jdk1.8.0 or above.		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Demonstrate the use of stream classes for reading and writing bytes/characters.

Note: Attach the code at the end.

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
public class IOStreamsExample {
  public static void main(String args[]) throws IOException {
    File file = new File("D:/myFile.txt");
     FileInputStream fis = new FileInputStream(file);
     byte bytes[] = new byte[(int) file.length()];
    fis.read(bytes);
     File out = new File("D:/CopyOfmyFile.txt");
     FileOutputStream outputStream = new FileOutputStream(out);
    outputStream.write(bytes);
     outputStream.flush();
     System.out.println("Data successfully written in the specified
file");
  }
}
```

XI. Result (Output of Code):

Data successfully written in the specified file

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Enlist java stream classes,file operations.
2. Explain InputStream and OutputStream classes along with methods.
3. Explain Reader and Writer stream classes.
4. Find the compile-time error in the program given below.

```
import java.io.*;
class FileOutput
{
        public static void main (String args[])
        {
                FileOutputStream out;
                PrintStream p;
                try
                {
                        Out=new FileOutputStream( );
                        p=new PrintStream(out);
                        p.println("This is written to a file");
                        p.close( );
                  }
catch(Exception e)
        {
                System.err.println("Error writing to a file);
         }
             }
}                              (Space for answer)
```


1. Java stream class are 2 types :-
   1. Byte Stream Classes
   2. Character Stream Classes

file operations :- 1. FileInputStream
2. FileOutputStream

3. Reader classes are similar to input streams, and writer classes are similar to output streams. Reader classes descend from the abstract Reader class, whereas the Writer classes descend from the abstract Writer class. Both readers and writers are divided into low-level and high-level classes.

2. a. InputStream : The InputStream class of the java.io package is an abstract superclass that represents an input stream of bytes.
Since InputStream is an abstract class, it is not useful by itself.
However, its subclasses can be used to read data.
Methods of InputStream :-
read() - reads one byte of data from the input stream
read(byte[] array) - reads bytes from the stream and stores in the specified array
available() - returns the number of bytes available in the input stream
mark() - marks the position in the input stream up to which data has been read
reset() - returns the control to the point in the stream where the mark was set
skips() - skips and discards the specified number of bytes from the input stream
close() - closes the input stream

b. Output Stream : The OutputStream class of the java.io package is an abstract superclass that represents an output stream of bytes.
Since OutputStream is an abstract class, it is not useful by itself. However, its subclasses can be used to write data.
Methods of OutputStream :-
write() - writes the specified byte to the output stream
write(byte[] array) - writes the bytes from the specified array to the output stream
flush() - forces to write all data present in output stream to the destination
close() - closes the output stream

XIII. Exercise:

1. Develop a program to copy characters from one file to another.
2. Develop a program to write bytes to a file.
3. Develop a program to display the content of file supplied as command line arguments.

(Space for answer)

1.
```
import java.io.*;
class Copy
{
public static void main(String args[])throws IOException
{
FileInputStream Fread =new FileInputStream("p1.txt");
FileOutputStream Fwrite=new FileOutputStream("p2.txt") ;
 System.out.println("File is Copied");
int c;
while((c=Fread.read())!=-1)
Fwrite.write((char)c);
Fread.close();
Fwrite.close();
}}
```

2.
```
import java.io.*;
public class Byte {
public static void main(String[] args)
{
String strFilePath = "D://p1.txt";
try
{
FileOutputStream fos = new FileOutputStream(strFilePath);
String strContent = " Raman ";
fos.write(strContent.getBytes());
fos.close();
System.out.println("Data is written into the file named as p1");
}
catch(FileNotFoundException ex)
{
System.out.println("FileNotFoundException : " + ex);
}
catch(IOException ioe)
{
System.out.println("IOException : " + ioe);
}}}
```

3.
```
import java.io.*;
class Copy
{
  public static void main(String args[])
 {
   try
   {
     FileInputStream fr=new FileInputStream(new File(args[0]));
     int i=0;
     while((i=fr.read())!=-1)
     {
       System.out.print((char)i);
     }
     fr.close();
   }
   catch(ArrayIndexOutOfBoundsException ex)
   {
    System.out.println(" C:\\Users\\Desktop\\File1.txt");
   }
   catch(IOException ex)
   {
    System.out.println("File Does Not Found in given Directory. ");
   }
  }
}
```

VIII. Resources required

Sr. No.	Name of Resource	Suggested Broad Specification	Quantity	Remark
1	Hardware: Computer System	Computer (i3-i5 preferable), RAM minimum 2 GB and onwards	As per batch size	For all Experiments
2	Operating system	Windows / Linux		
3	Software	jdk1.8.0 or above.		

IX. Resources used

S. No.	Name of Resource	Broad Specification	Qty	Remarks (If any)
1	Computer System with broad specifications	Computer (i3-i5) preferrable) RAM minimum 2GB and onwards	As per batch size	
2	Software	J.D.K. 1.8.0		
3	Any other resource used			

X. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Write installation directory path of your directory?
2. Write value of path environment variable?
3. List folders created after installation.
4. Main method is declared as static. Justify.
5. Program is named with class containing main method. Justify.

(Space for answer)

1] ⇒ c:\ program file (x86)\ Java \ JDK 1.8.0 - 60\ bin

2] ⇒ c:\ program file (x86) \ Java \ JDK 1.8.0 - 60\ bin

3] ⇒ i) bin
ii) conf
iii) include
iv) Jmods
v) legal
vi) lebs

4]

⇒ static is the modifier i.e keyword due to static main() method can be accessed without an object

Due to static only single copy of that member is created

Signature :->

```
public static void main (String args[])
{
    - - - - - // code
}
```

5]

⇒ All Java programs must have entry point which is alway the main() method whenever the code called it.

Main() method automatically executed firstly due to static it invokes without an object.

The main method appears in every program inside the class that is part of application.

But, if application is complecated containing multiple files.

It is common method which get execute called as generally Main() method.

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Write any program to check multiple conditions using if statement.

```
class Ladder
{
  public static void main (String args[])
  {
      int no = 0;
      if (no > 0)
      {
         System.out.println (" Number is positive ");
      }
      else
      {
         System.out.println ("Number is Negative") ;
      }
      elseif (no == 0)
      {
         system.out.println (" Number is zero ");
      }
  }
}
```


XI. Result (Output of Code):

Output :-

Number is Negative

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. List operators used in if conditional statement.
2. In if-else construct which part will be executed if condition is true.
3. State the condition when the else part will be executed with example.

4. Which of the following operator is used in if:
   a. Assignment operator (=) b. comparison operator (==)

(Space for answer)

1]

⇒ <

< → less than

> → greater than

<= → less than equal to

>= → greater than equal to

== → equal to equal to

2]

⇒ If condition is true then off course if conditional part is executed.

3]

⇒ e.g →

```
int num = -1;
if (num > 0)
{
    System.out.println (" Positive ");
}
else
{
    System.out.println (" Negative ");
}
```

Here, The number is -1 that is negative therefore condition is false.
Hence, else statement will be ~~eg~~ executed.

4]

⇒ b) Comparison Operator (==)

is used in if condition. →

XIII. Exercise:

1. Write output of code in the given space.

Sr. No.	Program Code	Output

1.	public class NestedIfExample {
	
	public static void main(String args[]){
	int num=70;
	if(num< 100){
	System.out.println("number is less than 100");
	if(num> 50){
	System.out.println("number is greater than 50");
	}
	}
	}
	}
		Number is less than 100
		
		Number is greater than 50.

2.	class IfStatement {
	public static void main(String[] args) {
	int number = 10;
	if (number > 0) {
	System.out.println("Number is positive.");
	}
	System.out.println("This statement is always executed.");
	}
	}
		Number is Positive
		
		This statement is always executed.

2. Write a program to make the use of logical operators.
3. Write a program to check no is even or odd.

(Space for Answer)

2]

```
class Logical_Operator
{
    public static void main (String args[])
    {
        int num1=1, num2 = 2, num3 = 3;
        boolean result;

        result = (num1 > num2) ||
                 (num3 > num1);

        System.out.println (result);
```

```
        result = (num1 > num2) &&
                 (num3 > num)
        System.out.println (Result);
    }
}
```

9]

⇒

```
class EvenOdd
{
    public static void main(String args[])
    {
        int no = 10;
        if (no % 2 == 0)
        {
            System.out.println ("Number is Even:"
        }
        else
        {
            System.out.println ("Number is Odd:"
        }
    }
}
```

Output :

Sunday

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. What will happen if break is not written for a case in switch case?
2. When default case is executed?
3. List datatypes allowed in switch expression?
4. Write a program to make use of ternary operator.

(Space for Answer)

1]

⇒ Switch cases are used to execute only specific case statement based on the switch expression

If we dont use break statement at the end of each case, then program will execute each case present in program it executes all the cases till end of switch case block which misbehaves the output.

which is not proper way to execute program

2]

⇒ When value passed to case did not match ~~can~~ any case clause.
then, default statement is executed

3]

⇒ A switch works with the byte, short, char, int etc primitive data types.

4]

⇒

```
Class TernaryOperate
{
    public static void main(String args[])
    {
        int a = 10;
        int b = 20;

        String result = a>b ?
                "a is greate";
                b is greate";

        System.out.println(result);

    }
}
```

Sr. No.	Program Code	Error/Output

1.	public class SwitchCaseExample1 {
	public static void main(String args[]){
	int num=2;
	switch(num+2)
	{
	case 1:
	System.out.println("Case1: Value is: "+num);
	case 2:
	System.out.println("Case2: Value is: "+num);
	case 3:
	System.out.println("Case3: Value is: "+num);
	default:
	System.out.println("Default: Value is: "+num);
	}
	}
	}
		Defalt:
		Value is 2

2.	public class Program {
	public static void main(String[] args) {
	int value = 100;
	switch (value) {
	case 100:
	System.out.println(true);
	break;
	case 100:
	System.out.println(true);
	break;
	}
	}
	}
		Error :-
		Duplicate
		case label
		Case 100;

2. Write any program to check switch-case statement using character datatype.

(Space for Answer)

```
class char
{
    public static void main (String args[])
    {
        char ch = 'b';
```

```
switch (ch)
{
    case 'a':
    System.out.println (" Case 1 ");
    break;

    Case 'b' :
    System.out.println (" Case 2 ");
    break ;

    Case 'x' :
    System.out.println (" Case 3 ) ;
    break ;

    Case 'y'
    system.out.println (" Case 4 ");
    break;

    Case default :
    System.out.println (" Default :);
    break ;
  }

}
}
```

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer i3 RAM 2GB	As per	
2	Software	JDK 1.8.0 or above	batch size	

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Develop a program to print command line argument using for loop.

```
class Command
{
  public static void main (String args[])
  {
      int i, num ;
      For ( i=0 ; i<=10 ; i++)
      {
        num = Integer.parseInt (args[i]);
        System.out.println (" num");
      }
  }
}
```


XI. Result (Output of Code):

Output :-

0 1 2 3 4 5 6 7 8 9 10

more such questions so as to ensure the achievement of identified CO.

1. When for loop will be terminated?
2. Can we write a for loop without initialization? If yes, give example.
3. Write a for loop to increment index variable by 2 in each iteration.
4. When for loop will be executed infinitely?

(Space for answer)

1]

⇒ When the break break statement is encountered inside a loop, the loop is immediately terminated.

2]

⇒ Yes, a for loop can be written without initialization

for loop statement usually goes like for (initialization ; test condition ; update)

3]

⇒

```
class Hello
{
    public static void main(String args[])
    {
        int value = 0;
        for(int i=0; i++)
        {
            if (i%2 == 0)
            {
                value++;
            }
        }
    }
}
```

4]

Here, a loop which will never stop or i.e never end is named to be infinite loop.

If the terminating condition is not given or if the condition cause the loop to start again then it is called as infinite loop.

XIII. Exercise:

1. Write any program using if condition with for loop.
2. Write any program to display pyramids of stars/patterns using increment/decrement

(Space for Answer)

1]

```
class check
{
public static void main(String args[])
{
for(i=1; i<=100; i++)
{
IF(i%2==0)
{
```

```
            System.out.println(i);
        }
    }
}
}
```

2]

⇒

```
class Pattern
{
    public static void main (String args[])
    {
        int rows = 5;

        for (int i=1; i<=rows; ++i)
        {
            for(int j=1; j<=i; ++j)
            {
                System.out.println("*");
            }
        }
    }
}
```

VIII. Resources required (Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	computer System	computer is RAM 2GB min	As per batch	
2	software	JDK 1.8.0 or above	size	

X. Program Code: Teacher must assign a separate program statement to group of 4 students.

Develop a program to use logical operators in do-while loop.

```
class Logic
{
   public static void main (String args[])
   {
      int i=1;
      while (i>=0) || (i<=10)
      {
         System.out.println (" line " +i);
         i++;
      }
   }
}
```


XI. Result (Output of Code):

output :

line 1 line 2 line 3
line 4 line 5 line 6
line 7 line 8 line 9 line 10

more such questions so as to ensure the achievement of identified CO.

1. State difference between while and do-while loop.
2. In do-while loop termination condition is checked at _____(beginning/end)
3. How many times do-while loop will be executed if condition is false?

(Space for answer)

1]

While	Do-While
i) while loop first check the condition then entee in the body	i) do while loop firstly entee in the loop then check the condition
ii) While loop is entry controlled loop	ii) do-while loop is exit controlled loop
iii) In while condition comes before body	iii) In do-while loop condition comes after body

2]

=) In do-while loop termination condition is checked at the end

3]

=) As we know do while loop first enters in the body then check the condition so, if condition though false the loop will be executed one's

XIII. Exercise:

1. Write Error/output of code in the given space.

Sr. No.	Program Code	Error/Output

1.	class DoWhileBasics
	{
	public static void main(String args[])
	{
	int a=1;
	do
	{
	System.out.println(a);
	a=a+1; // or a++;
	}
	while(a<=10);
	}
	}
		1
		2
		3
		4
		5
		6
		7
		8
		9
		10

2.	class Test {
	public static void main(String[] args)
	{
	while(true) {
	System.out.print(1);
	do {
	System.out.print(2);
	} while (false);
	}
	}
	}
		1212121212
		1212121212
		1212121212
		1212121212

2. https://www.youtube.com/watch?v=nX6cLed750
3. https://www.journaldev.com/16536/java-do-while-loop
4. https://beginnersbook.com/2015/03/do-while-loop-in-java-with-example/

(Space for answer)

2]

```
class Do
{
    public static void main (String args[])
    {
        int i=1 ;
        {
        System.out.println(i);
        i++;
        while(i<=50);
        }
    }
}
```

No.		
1.	byte	short, char, int, long, float, double
2.	short	int, long, float, double
3.	char	int, long, float, double
4.	long	float, double
5.	float	double

XII.

VIII. Resources required (Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	Computer i3 RAM 2GB	As per batch	
2	software	JDK 1.8.0 or above	size	

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Develop a program to show the use of implicit typecasting.

```
class Test
{
  public static void main (String args[])
  {
    int i = 100;
    long l = i;
    float f = l;
    System.out.println("Int value :" + i);
    System.out.println("Long valve:" + l);
    System.out.println("Float value :" + f);
  }
}
```

XI. Result (Output of Code):

Output :→
Int value : 100
Long value : 100
float value : 100.0

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. List different data types according to storage capacity.
2. State need of typecasting.
3. State the data types to which boolean datatype is implicitly casted.
4. Write two examples of implicit type casting.

(Space for answer)

1]
⇒ i) short
ii) Int
iii) Long
iv) float

2]
⇒ Type casting or type conversion is a method of changing an entity from one data type to another.
It is used in computer programming to ensure variable are correctly processed by function

3]

⇒ A boolean data type represent only one bit of information either true or false

Values of type boolean are not converted implicitly or explicitly (with cast) to any other type.

4]

⇒

```
class Implicit
{
public static void main (String args[])
{
byte i=10;
long j = i+8;
double k = i * 2.5+j ;

System.out.println("I ="+i);
System.out.println("J ="+J);
System.out.println("K ="+k);

}

}
```

1. Write Error/output of code in the given space.

Sr. No.	Program Code	Error/Output

1.	class Test{
	public static void main(String[] args) {
	int i = 100;
	long l = i;
	float f = l;
	System.out.println("Int value "+i);
	System.out.println("Long value "+l);
	System.out.println("Float value "+f);
	}
	}
		Output :
		Int value 100
		Long value 100
		Float value 100

2.	public class Test{
	public static void main(String[] argv) {
	char ch = 'c';
	int num = 88;
	ch = num;
	}
	}
		Error :-
		Possible lossy
		conversion from
		int to char
		ch = = num;

3. Write a program to implicitly typecast lower range data type to larger storage size datatype.

```
class Implicit
{
  public static void main (String args[]).
  {
      byte i = 50;
      long m = i;
      double n = m;
      System.out.println ("byte = " + i);
      System.out.println (" Long = " + m);
      System.out.println (" Double = " + m);
  }
}
```

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	computer i3 RAM 2GB	As per batch	
2	Software	JDK 1.8.0 or above	size	

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Develop a program to show the use of explicit type casting.

```
class Test
{
    public static void main (String args[])
    {
        double d = 100.04;
        long l = (long) d;
        int i = (int) d;
        System.out.println ("Double = " +d);
        System.out.println ("Long =" + l);
        System.out.println ("Int = " +i);
    }
}
```


XI. Result (Output of Code):

Output :

```
Double = 100.04
Long = 100
Int = 100
```

more such questions so as to ensure the achievement of identified CO.

1. What is casting?
2. What is difference between implicit and explicit type casting?
3. What is narrowing?

(Space for answer)

1]

⇒ Type casting in Java is to class one type of class or interface into another type i.e another also comes with the risk of class cast exception in Java which is quite common with a method which accepts object type and later type cast into more specific type

2]

Implicit	Explicit
i) ~~Implicit~~ implicit means done by Java virtual machine (JVM) or by tool	i) Explit means done by programmer i.e user
ii) This type of conversion happens automatically by the compiler	ii) In this type of conversion the programer deliberately converts the answer or variable in a specific data type of his choice

8]

⇒ The narrowing conversion occurs from a data type to a different type that has smaller size such as from a long (64 bits) to an (32 bits).

In general, narrowing primitive conversion can occur in these case short to byte or char.

XIII. Exercise:

1. Write Error/output of code in the given space.

Sr. No.	Program Code	Error/Output

1.	class Test{
	public static void main(String[] args) {
	double d = 100.04;
	long l = (long)d;
	int i = (int)l;
	System.out.println("Double value "+d);
	System.out.println("Long value "+l);
	System.out.println("Int value "+i);
	}
	}
		Double value 100.04
		Long value 100
		Int value 100

2.	class Test{
	public static void main(String args[]) {
	byte b = 50;
	b = (byte)(b * 2);
	System.out.println(b);
	}
	}
		output :
		100

3.	class Test{
	public static void main(String args[]) {
	byte a= 4;
	char b = 'z';
	short c = 102;
	int i = 5000;
	float f = 5.7f;
	double d = .124;
	double result = (f * a) + (i / b) - (d * c);
	System.out.println("result = " + result);
	}
	}
		output
		result =
		50.151999

2. Write a program to convert variable of basic datatypes and shows result of explicit typecasting.

(Space for Answer)

```
class Explicit
{
    public static void main (string args[])
    {
        byte x = 10;
        long y = (long) x;
        double z = (double) y;

        double result = x + y + z;

        System.out.println (" result);
    }
}
```

}

VIII. Resources required(Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	computer system	computer i3 RAM 2GB	As per batch	
2	software	JDK 1.8.0 or above	size	

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Demonstrate use of at least two types of constructors.

```
class Const
{
   public static void main(String args[]).
   {
      Const ()
      {
        System.out.println (" Default constructor:
      }
      Const (int a)
      {
        int x = a;
        System.out.println (x);
      }
}
```

XI. Result (Output of Code):

Output:

Default constructor 100

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Does constructor return a value?
2. Specify the situation when the default constructor is provided by the system.
3. Specify the situation when the default constructor is explicitly defined in the class.
4. How constructor overloading can be done?

(Space for answer)

1]

⇒ The constructor does not return a value is not a correct statement

According to me, it returns the current class instance but you cannot use return type yet it returns a value

i.e. current class instance

2]

⇒ When a variable of class type is set to null, it means that the variable was not initialized.

When we provide at least one constructor to our class, then the constructor compiler no more provides a constructor

3]

⇒ If constructors are explicitly defined for a class, but they are all non default the compiler will not implicitly define a default constructor. leading to a situation where the class does not have a default constructor

4]

⇒ Constructor can be overloaded in a similar way as function overloading

overloaded constructor can have the same name but different parameter

Depending upon the number and type of arguments passed, specific constructor called.

...given space.

Sr. No.	Program Code	Output

1.	class T {
	int t;
	}
	class Main {
	public static void main(String args[])
	{
	T t1 = new T();
	System.out.println(t1.t);
	}
	}
		Output :-
		
		0. (zero)

2. Modify the following program to execute without error. State which constructors are used in the program.

```
class Point
{
  int m_x, m_y;
  public Point(int x, int y)
  {  m_x = x;   m_y = y;  }
 public static void main(String args[])
 {
    Point p1 = new Point();
    Point p = new Point(2,3);
    System.out.println("X"+p.m_x);
    System.out.println("Y"+p.m_y);
    System.out.println("X"+p1.m_x);
    System.out.println("Y"+p1.m_y);
 }
}
```


3. Write a program to implement different types of constructors to perform addition of complex numbers.

(Space for Answer)

```
class const
{
    int a, b, sum;

    const ()
    {
        sum = 0;
    }
```

```
const(int x, int y)
{
a = x;
b = y;
sum = x + y;
System.out.println( sum);
}
}
```

XIV. References/ Suggestions for Further Reading

1. https://www.youtube.com/watch?v=lrYghXs9EEU
2. https://freevideolectures.com/course/2513/java-programming/17

XV. Assessment Scheme

Performance Indicators		Weightage
Process related (35 Marks)		**70%**
1	Logic formation	30%
2	Debugging ability	30%
3	Follow ethical practices	10%
Product related (15 Marks)		**30%**
4	Expected output	10%
5	Timely Submission	10%
6	Answer to sample questions	10%
Total (50 Marks)		**100%**

List of Students /Team Members

1. ... .
2. ...
3. ....
4. .....

Marks Obtained			Dated signature of Teacher
Process Related(35)	**Product Related(15)**	**Total(50)**	

7.	public boolean startsWith(String prefix)	prefix.
8.	public boolean endsWith(String suffix)	Checks if this string ends with given suffix.
9.	int indexOf(int ch)	A method returns index of given character value or substring.
10.	String substring(int startIndex)	Returns new string that is substring of this string
11.	int lastIndexOf(int ch)	A method returns last occurrence of given character value or substring.

String Buffer class methods:

Sr. No.	Syntax	Task Performed
1.	StringBuffer append (StringBuffer sb)	Appends specified StringBuffer with this StringBuffer
2.	StringBuffer insert (int offset, String str)	This method inserts a string str at position mentioned by offset.
3.	void setLength(int newlength)	Sets the length of the character sequence
4.	void setCharAt(int index, char ch)	The character at specified index of this StringBuffer is set to ch.
5.	StringBuffer reverse()	Reverse the character sequence in this StringBuffer.

VIII. Resources required(Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	computer System	computer i3 RAM 2GB	As per batch size	
2	Software	JDK 1.8.0 or above		

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Write a program to show the use of all methods of String class.

```
class Example
{
    public static void main (String args[])
    {
        string str1 = "My String";
        char arr [] = {'h','e','l','l','o'};
        string str2 = new string (arr);
        string str3 = new string ("Java string");
        System.out.println (str1);
        System.out.println (str2);
        System.out.println (str3);
    }
}
```

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Write a program to show the use of all methods of String class.

```
class Example
{
    public static void main(String args[])
    {
        string str1 = "My String";
        char arr[] = {'h','e','l','l','o'};
        string str2 = new string(arr);
        string str3 = new string("Java string");
        System.out.println(str1);
        System.out.println(str2);
        System.out.println(str3);
    }
}
```

XI. Result (Output of Code):

Output :-

My string

hello

Java string

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. List different constructors of String class along with syntax
2. List different constructors of StringBuffer class along with syntax.
3. State whether String is primitive datatype or class in Java? State the package.
4. What is difference between = = , equals() and compareTo() method?

(Space for answer)

1]

⇒ i) String ()

ii) String (byte [] byte)

iii) String (char [] value)

iv) String (String original)

v) String (StringBuffer buffer)

2]

⇒ String such as converting string instance and performang string concadenation, string is not a primitive type, but a class.

string is a class in Java and reference data type.

string is present in Java.lang package

4]

equals ( )	Compare ( )
i) This method is used to compare two strings	i) This method compares the string using directory order
ii) If two contents of string 'str' is same as invoking string then it returns true otherwise returns false	ii) A string is less than another if it comes before the other in dictionary order.

XIII. Exercise:

1. Write output of code in the given space.

Sr. No.	Program Code	Output

1.	class String_demo{
	public static void main(String args[]) {
	char chars[] = {'a','b','c'};
	String s = new String(chars);
	System.out.println(s);
	}
	}
		Output:
		
		abc

2.	class Output{
	public static void main(String args[]) {
	String s1 = "Hello I love Java" ;
	String s2 = new String(s1);
	System.out.println((s1==s2) + " " +
	s1.equals(s2));
	}
	}
		false true

2. Write a program to implement all methods of StringBuffer class.

2]

⇒

```
class StringBufferDemo
{
    public static void main(String args[])
    {
        StringBuffer sb = new StringBuffer
                          ("Hello");
        System.out.println("buffer before = "+s
        System.out.println("charAt(1) before
                          + sb.charAt(1
        sb.setchart(1,'i');
        sb.setlength(2);
        System.out.println("buffer after = "+sb);
        System.out.println("charAt(1) after="
                          + charAt(1));
    }
}
```

Output :-

buffer before = Hello
charAt(1) before = e
buffer after = Hi
charAt(1) after = i

```
int [] intArray = new int[20];
```

2. Multi Dimensional
   Example:

```
int[][] intArray = new int[10][20]; //a 2D array or matrix
int[][][] intArray = new int[10][20][10]; //a 3D array
```

VIII. Resources required(Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	computer i3 RAM 2GB	As per	
2	software	Java J.D.K 1.8.0	batch size	

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Write a program to implement multidimensional array.

```
class Multi
{
    public static void main (String args[])
    {
        int [][] a={{1,2,3},{4,5,6},{7}};
        System.out.println("length row1 :"+a[0].length);
        System.out.println("Length row2:"+a[1].length);
        System.out.println("length row3 :"+a[2].length);
    }
}
```

XI. Result (Output of Code):

output:-
length row1 : 3
length row2 : 3
length row3 : 1

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. What is use of new operator in defining an array?
2. In 2D array which dimension is optional at the declaration of array?
3. Is it possible to change size of array once allocated?
4. State the situation where Index Out of Bounds Exception will be generated.

(Space for answer)

1]

→ An array is an instance of a specral java array class and has a corresponding type in the type system.

This means that to use an array, as with other object, we first declare a variable

2]

→ In 2D array dimension the second dimension is optional.

3]

→ In Java array cannot be resized once it is declared

If you want a dynamic data structure with random access you use array

4]

If a request for a negative or an index greater than or equal to size of array is made, then the java throws a array / Index out of Bounds Exception

XIII. Exercise:

1. Write output/error of code in the given space.

Sr. No.	Program Code	Output/Error

1.	State line no and error.
	classTest2 {
	public static void main(String[] args) {
	inta[] = new int[5]; // line 1
	int[] arr = new int[]; // line 2
	}
	}
		Error :-
		array dimension missing
		int [] a11 = new int []

2.	class Test5 {
	public static void main(String[] args) {
	int arr[] = new int[5];
	System.out.println(arr);
	System.out.println(arr[0]);
	}
	}
		output:
		[I@ Idbj
		4720

3. Write a program to display array elements using for-each loop.

(Space for answer)

```
class For
{
    public static void main (String args[])
    {
        int a [] = { 10, 20, 30, 40, 50};
        for (int x = a)
        {
            System.out.println (x);
        }
    }
}
```

output :

10
20
30
40
50

XIV. References/ Suggestions for Further Reading

1. https://www.javatpoint.com/array-in-java
2. https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html
3. https://www.youtube.com/watch?v=okHL1h5rhNM

XV. Assessment Scheme

Performance Indicators		Weightage
Process related(35 Marks)		**70%**
1	Logic formation	30%
2	Debugging ability	30%
3	Follow ethical practices	10%
Product related (15 Marks)		**30%**
4	Expected output	10%
5	Timely Submission	10%
6	Answer to sample questions	10%
Total (50 Marks)		**100%**

List of Students /Team Members

1.
2. .....
3. ...
4. .....

Marks Obtained			Dated signature of Teacher
Process Related (35)	**Product Related (15)**	**Total (50**	

6.	Enumeration elements()	Returns enumeration of components of this vector
7.	Object firstElement()	Returns first component of this vector
8.	Object lastElement()	Returns last component of this vector
9.	int indexOf(Object elem)	Searches for the first occurrence of given argument.
10.	void insertElementAt(Object obj, int index)	Inserts specified object as a component at the specified index position.
11.	void removeElementAt(int index)	Removes the element at specified position in the vector
12.	boolean removeElement(Object obj)	Removes first occurrence of the argument from the vector
13.	int size()	Returns number of components in the vector
14.	void copyInto(Object[] array)	Copies the components of vector into specified array.

VIII. Resources required(Additional)

Nil

IX. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	computer i3 RAM 2GB	As per batch	
2	software	J.D.K 1.8.0	size	

X. Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Write a program to insert different elements in the Vector and display them.

```
class Test
{
  public static void main(String args[])
  {
    vector v = new vector();
    v.insertElement(new Integer(10));
    v.insertElement(new Integer(20));
    for(int i=0; i<v.size(); i++)
    {
      System.out.println(v.elementAt(i));
```

XI. Result (Output of Code):

Output :

10

20

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. State difference between size() and capacity() method of Vector class.
2. Differentiate between addElement() and insertElementAt() methods of Vector class.
3. Differentiate between array and Vector.

(Space for answer)

1]

Size()	Capacity()
i) Size() returns the number of element currently in the vector	i) Capacity() is the return capacity of the vector

2]

i) addElement()	ii) InsertElementAt()
The object specified by element is added to the end of vector	Add the element to the vector at the location specified index

Array	Vector
i) Array is fixed in size	vector is dynamic in size
ii) It holds value of primitive and object type	ii) It holds only object type value
iii) It is not class and does not contain any method	iii) vector is a class and contain lots of method

XIII. Exercise:

Sr. No.	Program Code	Output

1.	import java.util.*;
	class demo1 {
	public static void main(String[] args)
	{
	Vector v = new Vector(20);
	System.out.println(v.capacity());
	System.out.println(v.size());
	}
	}
		20
		0

1. Write a program to use different methods of Vector class.

(Space for answer)

```
class Vect
{
public static void main (String args[])
{
Vector v = new vector ();
v.addElement (10);
v.addElement (20);
v.addElement (30);
v.addElement (40);
```

```
v.removeElement At (2);
v.removeElement At (3);

for (int i=0; i< v.size(); i++)
{
    System.out.println( V.elementAt(i));
}
```

XIV. References/ Suggestions for Further Reading

1. https://www.javatpoint.com/array-in-java
2. https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html
3. https://www.youtube.com/watch?v=okHL1h5rhNM

XV. Assessment Scheme

Performance Indicators		Weightage
Process related(35 Marks)		**70%**
1	Logic formation	30%
2	Debugging ability	30%
3	Follow ethical practices	10%
Product related (15 Marks)		**30%**
4	Expected output	10%
5	Timely Submission	10%
6	Answer to sample questions	10%
Total (50 Marks)		**100%**

List of Students /Team Members

1. ....
2. ....
3. .....
4. .....

Marks Obtained			Dated signature of Teacher
Process Related (35)	**Product Related (15)**	**Total (50)**	

		has a greater value.
10	static int compare(int num1, int num2)	Compares the values of num1 and num2. Returns 0 if the values are equal. Returns a negative value if num1 is less than num2. Returns a positive value if num1 is greater than num2.
11	boolean equals(Object intObj)	Returns true if the invoking Integer object is equivalent to intObj. Otherwise, it returns false.

Similar Wrapper class methods are available for Float, Short, Long and Double class.

III. Resources required(Additional)

Nil

X. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specification	Quantity	Remarks (If any)
1	Computer System	computer 1g RAM 2GB	As per	
2	software	JDK 1.8.0 or above	batch size	

Program Code: Teacher must assign a separate program statement to group of 3-4 students.

Write a program to show the use of Integer Wrapper class methods.

```
import java.io.*;
class Binary
{
  public static void main(String args[]) throws
                                    IOException
  {
    BufferedReader br = new BufferedReader
                (new InputStreamReader
                (System.in));
    System.out.println("Enter no:");
    int i = Integer.parseInt(br.readLine());
    String binary = Integer.toBinaryString(i));
    System.out.println("Binary value:" +binary);
```

XI. Result (Output of Code):

Output :-

Enter number : 5

Binary value : 101

XII. Practical Related Questions

Note: Below given are few sample questions for reference. Teacher must design more such questions so as to ensure the achievement of identified CO.

1. Write a different ways to create object of the any primitive datatype.
2. Write methods of Number class to convert object into primitive datatypes.
3. List all Wrapper classes in Java.

(Space for answer)

1]

⇒ byte byte value ()
double double value ()
Float Float value ()
int int value
long long value
short short value

2]

⇒ byte – Return object value as byte
double – Return object value as double
Float – Return object value as float
int – Return object value as long
long – Return object value as long
short – Return object value as short

3]

i) Byte
ii) short
iii) character
iv) Integer
v) Float
vi) Long
vii) Double
viii) Boolean

XIII. Exercise:

1. Write a program to convert String value into Integer Wrapper class object.
2. Write a program to make use of Character Wrapper class methods.
3. Write a program to convert Integer object value into primitive datatype byte, short and double value.

(Space for answer)

1]

```
class string
{
    public static void main(String args[])
    {
        Integer obj1 = new Integer("100");
        System.out.println("int obj1");
        string str = "100";
        Integer.int obj2 = Integer.valueOf(str);
        System.out.println(intobj2);
    }
}
```

Output :-

100

100

XIV. References/ Suggestions for Further Reading

1. https://www.youtube.com/watch?v=IM3c6eI6lO8
2. https://www.tutorialspoint.com/java/java_numbers.htm

XV. Assessment Scheme

Performance Indicators		Weightage
Process related(35 Marks)		**70%**
1.	Logic formation	30%
2.	Debugging ability	30%
3.	Follow ethical practices	10%
Product related (15 Marks)		**30%**
4.	Expected output	10%
5.	Timely Submission	10%
6.	Answer to sample questions	10%
Total (50 Marks)		**100%**

List of Students /Team Members

1. ..
2. ..
3. .....
4. .....

Marks Obtained			Dated signature of Teacher
Process Related (35)	**Product Related (15)**	**Total(50)**	