WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Object Oriented Programming with C++ **Subject Code:** 22316

Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills).
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.

Q. No.	Sub Q.N.	Answer	Marking Scheme
1.	**a) Ans.**	**Attempt any FIVE of the following:** **State any four object oriented languages.** **Object oriented programming language:** • C++ • Smalltalk • Object pascal • java • Simula • Ada • Turbo pascal • Eiffel • C# • Python	**10** **2M** ***Any 4 languages ½ M each***
	b) Ans.	**Describe use of protected access specifier used in the class.** Protected access specifier is use to declare a class member that is accessible by the member functions within its class and any class immediately derived from it.	**2M** ***Correct use 2M***

c) Ans — **Differentiate between OOP and POP** — **2M**

Sr. No.	PROCEDURE ORIENTED PROGRAMMING (POP)	OBJECT ORIENTED PROGRAMMING (OOP)
1	Focus is on doing things (procedure).	Focus is on data rather than procedure.
2	Large programs are divided into multiple functions.	Programs are divided into multiple objects.
3	Data move openly around the system from function to function.	Data is hidden and cannot be accessed by external functions.
4	Functions transform data from one form to another by calling each other.	Objects communicate with each other through function.
5	Employs top-down approach in program design.	Employs bottom-up approach in program design
6	Procedure oriented approach is used in C language.	Object oriented approach is used in C++ language.

Any two relevant differences 1M each

d) Ans. — **Write any two characteristics of destructor.** — **2M**

Characteristics:
1. It is used to destroy objects created by a constructor.
2. Name of destructor and name of the class is same.
3. Its name is preceded with tilde (~) symbol.
4. It never takes any argument.
5. It does not return any value.
6. It is invoked implicitly by the compiler upon exit from the program (or block or function) i.e when scope of object is over.

Any two characteristics- 1M each

e) Ans. — **Describe meaning of the following** — **2M**

(i) ios : : in
(ii) ios : : out

(i) ios : : in : It is a file mode. It is used to open a file in read only mode.

(ii) ios : : out : It is a file mode. It is used to open a file in write only mode.

Meaning of 'in' 1M
Meaning of 'out' 1M

WINTER – 2018 EXAMINATION
MODEL ANSWER

Subject: Object Oriented Programming with C++ **Subject Code:** 22316

	f)	**Give output for following code:** **class student** **{** **int roll no;** **char name [14];** **} s[6];** **void main()** **{** **cout<<sizeof(s);** **}**	**2M**
	Ans	Considering roll_no(Single variable) the output is: 96 OR Considering roll, no (Two variables) the output is: 108 OR Considering roll no the output is: error – space between roll and no	***Correct output 2M***

	g)
	Ans
		Write syntax to define a derived class
		Syntax:
		class derived_class_name : visibility_mode/access_specifier base_class_name
		{
		class body
		};
			2M
			Correct syntax 2M

2		**Attempt any <u>THREE</u> of the following**	**12**
	a)	**Write a C++ program to accept array of five elements, find and display smallest number from an array.**	**4M**

	Ans	#include<iostream.h>
		#include<conio.h>
		void main()
		{
		int a[5],smallest,i;
		clrscr();
		cout<<" Enter array elements:";
		for(i=0;i<5;i++)
		cin>>a[i];
		smallest=a[0];
		for(i=1;i<5;i++)
		{
		if(a[i]<smallest)
			Correct logic 2M
			Correct syntax 2M

		{
		smallest=a[i];
		}
		}
		cout<<endl<<"Smallest number="<<smallest;
		getch();
		}	

	b)
	
	Ans
		Write a C++ program to declare a class 'College' with data members as name and college code. Derive a new class 'student' from the class college with data members as sname and roll no. Accept and display details of one student with college data.
		#include<iostream.h>
		#include<conio.h>
		class college
		{
		char name[10];
		int collegecode;
		public:
		void getcollege()
		{
		cout<<"Enter college name:";
		cin>>name;
		cout<<"Enter college code:";
		cin>>collegecode;
		}
		void putcollege()
		{
		cout<<endl<<"College name="<<name;
		cout<<endl<<"College code="<<collegecode;
		}
		};
		class student:public college
		{
		char sname[10];
		int rollno;
		public:
		void getstudent()
		{
		cout<<"Enter student name";
			4M
			
			Declaration and Definition of Base Class 1M
			
			Declaration and Definition of Derived Class 2M
			
			Main function 1M

		cin>>sname;
		cout<<"Enter roll no:";
		cin>>rollno;
		}
		void putstudent()
		{
		cout<<endl<<"Student name:="<<sname;
		cout<<endl<<"Roll no:="<<rollno;
		}
		};
		void main()
		{
		student s;
		clrscr();
		s.getcollege();
		s.getstudent();
		s.putcollege();
		s.putstudent();
		getch();
		}	

	c)
	
	Ans
		Write a C++ program to declare a class 'circle' with data members as radius and area. Declare a function getdata to accept radius and putdata to calculate and display area of circle.
		#include<iostream.h>
		#include<conio.h>
		class circle
		{
		float radius,area;
		public:
		void getdata()
		{
		cout<<"Enter radius:";
		cin>>radius;
		}
		void putdata()
		{
		area=3.14*radius*radius;
		cout<<"Area of circle="<<area;
			4M
			
			Decalaration and Definition of class with functions 3M

		}
		};
		void main()
		{
		circle c;
		clrscr();
		c.getdata();
		c.putdata();
		getch();
		}
			Main function 1M

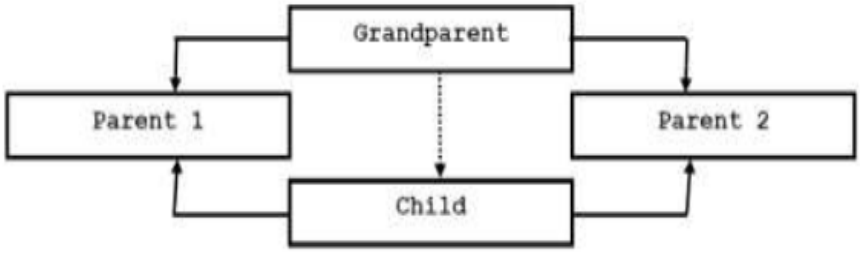
	d)
	
	Ans.
		With suitable example, describe effect of ++ and - - operators used with pointer in pointer arithmetic.
		++ Operator: - It is referred as increment operator that increments the value of variable. If ++ operator is used with pointer variable, then pointer variable points to next memory address that means pointer increment with respect to size of the data type used to declare pointer variable.
		
		Example:-
		int a[5]={10,20,30,40,50},*ptr;
		ptr=a[0];
		for(i=0;i<5;i++)
		{
		cout<<*ptr;
		ptr++;
		}
		In the above example, ptr points to memory location of a[0]. Increment statement ptr++ increments ptr by memory size of int i.e 2 bytes and ptr points to a[1].
		
		- - Operator: - It is referred as decrement operator that decrements the value of variable. If - - operator is used with pointer variable, then pointer variable points to previous memory address that means pointer decrement with respect to size of the data type used to declare pointer variable.
			4M
			
			Description of ++ operator 1M
			
			Any relevant Example 1M
			
			Description of - - operator 1M

		Example:-
		int a[5]={10,20,30,40,50},*ptr;
		ptr=a[4];
		for(i=0;i<5;i++)
		{
		cout<<*ptr;
		ptr- -;
		}
		
		In the above example, ptr points to memory location of a[4]. Decrement statement ptr- - decrements ptr by memory size of int i.e 2 bytes and ptr points to a[3].
			Example 1M

3	**a)**
	
	Ans.
		Attempt any <u>THREE</u> of the following
		Write a C++ program to declare a class addition with data members as x and y. Initialize values of x and y with constructor. Calculate addition and display it using function 'display'.
		#include<iostream.h>
		#include<conio.h>
		class addition
		{
		int x,y;
		public:
		addition(int,int);
		void display();
		};
		addition::addition (int x1,int y1)
		{
		x=x1;
		y=y1;
		}
		void addition::display()
		{
		cout<<"\nAddition of two numbers is:"<<(x+y);
		}
		void main()
		{
		addition a(3,4);
			12
			4M
			
			Declaration and definition of class with constructor and display function 3M
			
			Main function 1M

		a.display();
		getch();
		}	

	b) **Ans**	**With suitable diagram describe structure of C++ program.** General C++ program has following structure. INCLUDE HEADER FILES CLASS DECLARATION MEMBER FUNCTIONS DEFINITIONS MAIN FUNCTION PROGRAM **Description:-** **1. Include header files** In this section a programmer include all header files which are require to execute given program. The most important file is *iostream.h* header file. This file defines most of the C++statements like *cout* and *cin*. Without this file one cannot load C++ program. **2. Class Declaration** In this section a programmer declares all classes which are necessary for given program. The programmer uses general syntax of creating class. **3. Member Functions Definition** This section allows programmer to design member functions of a class. The programmer can have inside declaration of a function or outside declaration of a function. **4. Main Function Program** In this section programmer creates objects and calls various functions writer within various class.	**4M** ***Correct diagram 2M*** ***Description 2M***
	c) **Ans.**	**Describe the concept of virtual base class with suitable example.** ***Note: Program/diagram with syntax shall be considered as an example.*** **Virtual Base Class:** An ancestor class is declared as virtual base class which is used to avoid duplication of inherited members inside child class due to multiple path of inheritance.	**4M** ***Description 2M***



Consider a hybrid inheritance as shown in the above diagram. The child class has two direct base classes, Parent1 and Parent2 which themselves have a common base class as Grandparent. The child inherits the members of Grandparent via two separate paths. All the public and protected members of Grandparent are inherited into Child twice, first via Parent1 and again via Parent2. This leads to duplicate sets of the inherited members of Grandparent inside Child class. The duplication of inherited members can be avoided by making the common base class as virtual base class while declaring the direct or intermediate base classes as shown below.

```
class Grandparent
{
};
class Parent1:virtual public Grandparent
{
};
class Parent2:virtual public Grandparent
{
};
class Child: public Parent1,public Parent2
{
};
```

Example 2M

Example

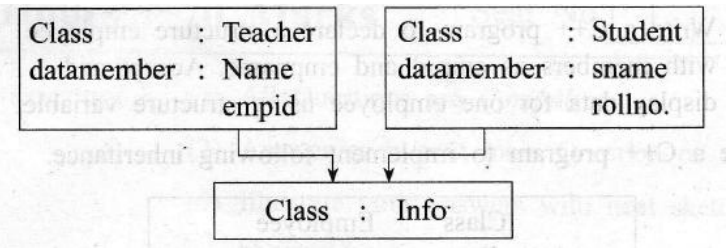
```
#include<iostream.h>
#include<conio.h>
class student
{
int rno;
```

```
public:
void getnumber()
{
cout<<"Enter Roll No:";
cin>>rno;
}
void putnumber()
{
cout<<"\n\n\t Roll No:"<<rno<<"\n";
}
};
class test: virtual public student
{
public:
int part1,part2;
void getmarks()
{
cout<<"Enter Marks\n";
cout<<"Part1:";
cin>>part1; cout<<"Part2:";
cin>>part2;
}
void putmarks()
{
cout<<"\t Marks Obtained\n";
cout<<"\n\t Part1:"<<part1;
cout<<"\n\tPart2:"<<part2;
}
};
class sports: public virtual student
{
public:
int score;
void getscore()
{
cout<<"Enter Sports Score:";
cin>>score;
}
void putscore()
```

		{
		cout<<"\n\t Sports Score is:"<<score;
		}
		};
		class result: public test, public sports
		{
		int total;
		public:
		void display()
		{
		total=part1+part2+score;
		putnumber();
		putmarks();
		putscore();
		cout<<"\n\t Total Score:"<<total;
		}
		};
		void main()
		{
		result obj;
		clrscr();
		obj.getnumber();
		obj.getmarks();
		obj.getscore();
		obj.display();
		getch();
		}	

	d) Ans	**Describe use of static data member in C++ with example.** **Use of static data member**: Static data member is used to maintain values common to the entire class. It is initialized to zero when the first object of its class is created. Only one copy of that member is created for the entire class and is shared by all the objects of that class. **Example:** #include<iostream.h> #include<conio.h> class test {	**4M** ***Use of static data member 2M*** ***Relevant example 2M***

```
static int count;
int obj_no;
public:
void getdata()
{
obj_no=++count;
cout<<"\n Object number="<<obj_no;
}
static void showcount()
{
cout<<"\n total number of objects="<<count;
}
};
int test::count;
void main()
{
test t1,t2;
clrscr();
t1.getdata();
t2.getdata();
test::showcount();
test t3;
t3.getdata();
test::showcount();
getch();
}
```

4	**a)**	**Attempt any <u>THREE</u> of the following** **Write a C++ program to implement inheritance shown in following figure:**  **Accept and display data of one teacher and one student using object of class 'Info'** ***Note: Any other correct logic of multiple inheritance in program shall be considered.***	**12** **4M**

	Ans	#include<iostream.h>
		#include<conio.h>
		class Teacher
		{
		protected:
		char Name[20];
		int empid;
		};
		class Student
		{
		protected:
		char sname[20];
		int rollno;
		};
		class Info:public Teacher,public Student
		{
		public:
		void acceptT()
		{
		cout<<"\nEnter data for teacher:";
		cout<<"\nName:";
		cin>>Name;
		cout<<"\nEmployee id:";
		cin>>empid;
		}
		
		void displayT()
		{
		cout<<"\nTeacher's data is:";
		cout<<"\nName:"<<Name;
		cout<<"\nEmployee id:"<<empid;
		}
		void acceptS()
		{
		cout<<"\nEnter student's data:";
		cout<<"\nName:";
		cin>>sname;
			Correct definition of class - Teacher 1M
			
			Correct definition of class- Student 1M
			
			Correct definition of class- Info 1M

		cout<<"\nRoll no:";
		cin>>rollno;
		}
		void displayS()
		{
		cout<<"\nStudent's data is:";
		cout<<"\nName:"<<sname;
		cout<<"\nRoll no:"<<rollno;
		}
		};
		void main()
		{
		Info I;
		clrscr();
		I.acceptT();
		I.displayT();
		I.acceptS();
		I.displayS();
		getch();
		}
			Correct definition of main function 1M

	b)
	
	Ans
		Write a C++ program to print multiplication table of 7. (example: 7 x 1 ….7 x 10 = 70)
		#include<iostream.h>
		#include<conio.h>
		void main()
		{
		int num;
		clrscr();
		cout<<"Multiplication table for 7 is:"<<endl;
		for(num=1;num<=10;num++)
		{
		cout<<"7 *"<<num<<"="<<7*num<<endl;
		}
		getch();
		}
			4M
			
			Correct logic 2M
			
			Correct syntax 2M

	c)	**Write a C++ program to swap two integer numbers and swap two float numbers using function overloading.**	**4M**

		(Hint: overload swap function) ***Note: Any other relevant logic shall be considered.***	

	Ans	#include<iostream.h>
		#include<conio.h>
		void swap(int a,int b)
		{
		int temp;
		temp=a;
		a=b;
		b=temp;
		cout<<"\nInteger values after swapping are:"<<a<<" "<<b;
		}
		void swap(float x,float y)
		{
		float temp1=x;
		x=y;
		y=temp1;
		cout<<"\nFloat values after swapping are:"<<x<<" "<<y;
		}
		void main()
		{
		clrscr();
		swap(10,20);
		swap(10.15f,20.25f);
		getch();
		}
			Correct logic 2M
			
			Correct syntax 2M

	d)	**Write a C++ program to count number of spaces present in contents of file.** ***Note: Any other relevant logic shall be considered***	**4M**
	Ans	#include<iostream.h> #include<fstream.h> #include<conio.h> void main() { ifstream file; charch;	***Correct logic 2M*** ***Correct***

		int s=0;
		clrscr();
		file.open("abc.txt");
		while(file)
		{
		file.get(ch);
		if(ch==' ')
		{
		s++;
		}
		}
		cout<<"\nNumber of spaces present in the content of the given file are:"<<s;
		getch();
		}
			syntax 2M

	e)
	
	Ans.
		Write a C++ program to find greatest number among two numbers from two different classes using friend function.
		#include<iostream.h>
		#include<conio.h>
		class second;
		class first
		{
		int x;
		public:
		void getx()
		{
		cout<<"\nEnter the value of x:";
		cin>>x;
		}
		friend void max(first,second);
		};
		class second
		{
		int y;
		public:
		void gety()
		{
		cout<<"\nEnter the value of y:";
			4M
			
			Correct definition of class first 1M
			
			Correct definition of class second

		cin>>y;
		}
		friend void max(first,second);
		};
		void max(first a,second b)
		{
		if(a.x>b.y)
		{
		cout<<"\Greater value is:"<<a.x;
		}
		else
		{
		cout<<"\nGreater value is:"<<b.y;
		}
		}
		void main()
		{
		first a;
		second b;
		clrscr();
		a.getx();
		b.gety();
		max(a,b);
		getch();
		}
			1M
			
			Correct definition of friend function 1M ,
			
			Correct definition of main function 1M

5	**a)** **Ans**	**Attempt any <u>TWO</u> of the following** **Write a C++ program to overload binary operator '+' to concatenate two strings.** #include<iostream.h> #include<conio.h> #include<string.h> class opov { char str1[10]; public: void getdata() {	**12** **6M** ***Creating Class 2M*** ***Operator Functio***

		cout<<"\nEnter a strings";
		cin>>str1;
		}
		void operator +(opov o)
		{
		cout<<strcat(str1,o.str1);
		}
		};
		void main()
		{
		opov o1,o2;
		clrscr();
		o1.getdata();
		o2.getdata();
		o1+o2;
		getch();
		}
			n
			2M
			
			Main Functio n
			2M

	b)
	
	Ans
		Write a C++ program to write 'Welcome to poly' in a file. Then read the data from file and display it on screen.
		Note: Any other relevant logic shall be considered
		#include<iostream.h>
		#include<conio.h>
		#include<fstream.h>
		void main()
		{
		char str[25] = "Welcome to poly",ch;
		clrscr();
		ofstream fout;
		fout.open("output.txt");
		fout<<str;
		fout.close();
		ifstream fin;
		fin.open("output.txt");
		while (!fin.eof())
		{
		fin.getline(str, 25);
		cout<<str<<endl;
		}
			6M
			
			Writing data in file 3M
			
			Reading data from file and display on screen 3M

		fin.close();
		getch();
		}	

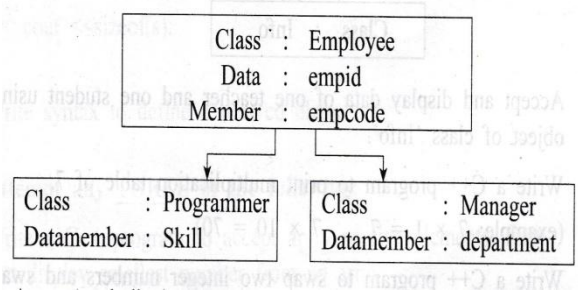
	c)
	
	Ans
		Write a C++ program to declare a class 'Account' with data members as accno, name and bal. Accept data for eight accounts and display details of accounts having balance less than 10,000.
		#include<iostream.h>
		#include<conio.h>
		class Account
		{
		long int accno, bal;
		char name[10];
		public:
		void getdata()
		{
		cout<<"\nEnter account number, balance and name ";
		cin>>accno>>bal>>name;
		}
		void putdata()
		{
		if(bal>10000)
		{
		cout<<"\nThe Account Number is "<<accno;
		cout<<"\nThe Balance is "<<bal;
		cout<<"\nThe Name is "<<name;
		}
		}
		};
		void main()
		{
		Account a[8];
		int i;
		clrscr();
		for(i=0;i<8;i++)
		{
		a[i].getdata();
		}
		for(i=0;i<8;i++)
		{
			6M
			
			Creating Class 2M
			
			Logic to Display object with given conditio n 1M
			
			Creating 8 objects 1M
			
			Calling function s 2M

		a[i].putdata();
		}
		getch();
		}	

6	**a)**	**Attempt any TWO of the following** **(i) Write a C++ program to find whether the entered number is even or odd.** **(ii) Write a C++ program to declare a structure employee with members as empid and empname. Accept and display data for one employee using structure variable.**	**12** **6M**

	Ans	**(i) Write a C++ program to find whether the entered number is even or odd.**
		
		#include<iostream.h>
		#include<conio.h>
		void main()
		{
		int num;
		clrscr();
		cout<<"\nEnter a Number ";
		cin>>num;
		if(num%2==0)
		{
		cout<<"\nEntered number is even";
		}
		else
		{
		cout<<"\nEntered number is odd";
		}
		getch();
		}
		
		(ii) Write a C++ program to declare a structure employee with members as empid and empname. Accept and display data for one employee using structure variable.
		
		#include<iostream.h>
		#include<conio.h>
			Accepting Number 1M
			
			Condition to check number 1M
			
			Display result 1M
			
			Creating structur

		struct employee
		{
		int empid;
		char empname[10];
		};
		void main()
		{
		employee e;
		clrscr();
		cout<<"\nEnter employee id and Employee Name ";
		cin>>e.empid>>e.empname;
		cout<<"\mThe Employee Id is "<<e.empid;
		cout<<"\nThe Employee Name is "<<e.empname;
		getch();
		}
			e with specified member 1M
			
			Accepting and displaying values 2M

	b)	**Write a C++ program to implement following inheritance.**  **Accept and display data for one programmer and one manager. Make display function virtual.**	**6M**
	Ans.	#include<iostream.h> #include<conio.h> class Employee { int empid,empcode; public: void emp() { cout<<"\nEnter an employee id "; cin>>empid; cout<<"\nEnter an employee code "; cin>>empcode;	***Creating all classes 3M***

```
        }
void virtual display()
        {
        cout<<"\nEmployee id "<<empid;
        cout<<"\nEmployee code"<<empcode;
        }
 };
class Programmer : public Employee
 {
char Skill[10];
public:
void getskill()
        {
        cout<<"\nEnter a Skill for Programmer ";
        cin>>Skill;
        }
void display()
        {
        cout<<"\nThe Programmer Skill is "<<Skill;
        }
 };
class Manager : public Employee
 {
char department[10];
public:
void getdept()
        {
        cout<<"\nEnter a Department for Manager ";
        cin>>department;
        }
void display()
        {
        cout<<"\nThe Department of Manager is "<<department;
        }
 };
void main()
 {
  Employee e, *eptr;
  Programmer p;
```

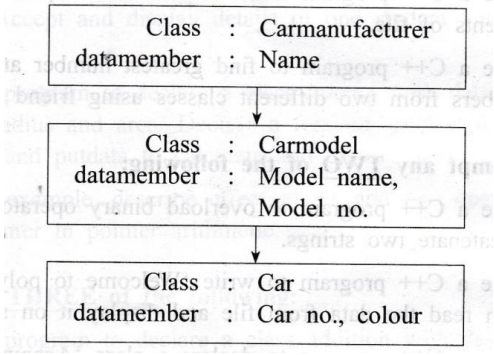
Main Function 3M

```
   Manager m;
clrscr();
cout<<"\nFor Programmer Class ";
eptr = &e;
eptr->emp();
p.getskill();
eptr->display();
eptr= &p;
eptr->display();

cout<<"\nFor Manager Class ";
eptr = &e;
eptr->emp();
m.getdept();
eptr->display();
eptr= &m;
eptr->display();

getch();
  }
```

c) **Write a C++ program for following multilevel inheritance.** **6M**



Accept and display data for one car with all details.

Ans

```
#include<iostream.h>
#include<conio.h>
class Carmanufacturer
  {
char Name[10];
```

Declarat

		public:
		void getcarm()
		{
		cout<<"\nEnter Car Name ";
		cin>>Name;
		}
		
		void putcarm()
		{
		cout<<"\nThe Car Name is "<<Name;
		}
		};
		class Carmodel : public Carmanufacturer
		{
		char Modelname[10];
		int Modelno;
		public:
		void getcarmodel()
		{
		cout<<"\nEnter Car Model Name and Model No. ";
		cin>>Modelname>>Modelno;
		}
		void putcarmodel()
		{
		cout<<"\nEnter Car Model Name and Model No.
		"<<Modelname<<" "<<Modelno;
		}
		};
		class Car: public Carmodel
		{
		char colour[10], Carno[10];
		public:
		void getcar()
		{
		cout<<"\nEnter Car colour and car number";
		cin>>colour>>Carno;
		}
		void putcar()
		{
			ion & Definitio n of all classes 3M

		cout<<"\nEnter Car colour and car number "<<colour<<" "<<Carno;
		}
		};
		void main()
		{
		Car c;
		clrscr();
		c.getcarm();
		c.getcarmodel();
		c.getcar();
		c.putcarm();
		c.putcarmodel();
		c.putcar();
		getch();
		}
			Main function 3M