CHAPTER-5
Constructors and Destructors
SHORT ANSWER QUESTIONS

1. **Differentiate between Constructor and Destructor function with respect to Object Oriented Programming.**

Ans.

Constructor	Destructor
Constructor is used to initialize the object.	Destructor is used to destroy the objects that are created in memory previously.
Constructor can takes arguments.	Destructor cannot take any arguments.
Constructor overloading can be possible means more than one constructor can be defined in same class.	Destructor overloading cannot be possible.

Syntax of constructor:
`class class_name`
`{`
`class_sname(){}`
`class_name(argulist){}`
`} ;`
	Syntax of destructor:
	`class class_name`
	`{`
	`~class-name(void){}`
	`};`

2. **When a compiler can automatically generate a constructor if it is not defined then why is it considered that writing constructor for a class is a good practice?**

Ans. When an object of the class is created a compiler can automatically generates a constructor if it is not defined. It is considered that writing constructor for a class is a good practice because constructor takes over very important duty of initialization of an object being created and relieves us from this task.

3. **'Accessibility of a constructor or a destructor greatly affects the scope and visibility of their class'. Elaborate this statement.**

Ans. Generally, a constructor and destructor should be defined under the public section of a class, so that its objects can be created and destroyed in any function. A private or protected constructor/destructor is not available to the non-member functions. Thus, accessibility of a constructor or a destructor greatly affects the scope and visibility of their class.

4. **Explain the role of a default constructor? When is it considered equivalent to a parameterized constructor? Support your answer with examples.**

Ans. A default constructor is the one that takes no argument. It is automatically invoked when an object is created without providing any initial values. In case, the programmer has not defined a default constructor, the compiler automatically generates it. For example,

```
class A {..........};
A ob1;   // uses default constructor for creating ob1.
```

A parameterized constructor with default argument is equivalent to a default constructor. For example,

```
class A {   int i;
            float j;
         public:
           A(int a=0,float b=1000.0);  //constructor with default argument
         };
A::A(int a,float b)     //constructor definition
{
     i=a; j=b;
}
int main()
{    A o1(23,27.50);   // argument value passed for o1
     A o2;   // takes default argument to o2(0,1000.0)
}
```

5. **List some of the special properties of the constructor functions.**

Ans.
- ✓ Constructor functions are invoked automatically when the objects are created.
- ✓ No return type can be specified for a constructor.

- ✓ A constructor may not be static.
- ✓ They cannot be inherited, though a derived class can call the base class constructor.
- ✓ It is not possible to take the address of a constructors.

6. **What is a parameterized constructor? How is it useful?**

Ans. A constructor that accepts parameters for its invocation is known as parameterized constructor. This helps you to assign initial value to an object at the time of its creation. For example,

```
class Test  {   int ant;
              public:
                Test(int i)    // constructor with argument
                {    ant=i; }
             };
int main()
{    Test ob1(45);     //argument value provided
}
```

7. **What is a copy constructor? What is its significance? Which situation is it invoked in? Support your answer with examples.**

Ans. The copy constructor is a constructor which creates an object by initializing it with an object of the same class, which has been created previously. The copy constructor is used to:
Initialize one object from another of the same type.
Copy an object to pass it as an argument to a function.
Copy an object to return it from a function.

```
#include<iostream>
#include<conio.h>
class Example{
    int a,b;
    public:
       Example(int x,int y){         //Constructor with Argument
           a=x;
           b=y;
          cout<<"\nParameterized Constructor";
      }
      void Display(){
          cout<<"\nValues :"<<a<<"\t"<<b;
      }
};
void main(){
       Example Object(10,20);
       Example Object2=Object;        //Copy Constructor
       Object.Display();             // Constructor invoked.
       Object2.Display();
        getch();
}
```

8. **Differentiate between a default constructor and copy constructor, giving suitable examples of each.**

Ans.

Default Constructor	**Copy Constructor**
A constructor that accepts no parameter is called the default constructor.	A constructor that initializes a object with the data values of another object is called copy constructor.
A default constructor takes no parameter.	Copy constructor takes one parameter of its class& type.

Example:
`class Defal`
`{    public:`
`Defal()`
`{  cout<<" Default constructor";   }`
	Example:
	`class A {  int i;`
	`public:`
	`A(int a)  //constructor`
	`{   i=a;   }`
	`A(A &s)//copy constructor`

	`};`	`{ i=s.i; }` `};`

9.	**Describe the importance of destructor. List some of the special properties of destructor.**
Ans.	A object that is existing must be scrapped off when it is o more needed. The task of scrapping off an object is carried out by a destructor. A destructor deinitializes an object and deallocates all allocated resources. Properties of destructor: ✓ Destructor functions are invoked automatically when the objects are destroyed. ✓ There can be only one destructor for a class, means destructor can't be overloaded. ✓ No argument can be provided to a destructor, neither does it returns any value.

10.	**What will be the output of following program? Explain with reasons:**
	`#include<iostream.h>`
	`class student{`
	`int rollno;`
	`char grade;`
	`static int count;`
	`public:`
	`student()`
	`{`
	`rollno=0; grade=' ';`
	`cout<<"Creating object"<<++count<<"\n";`
	`}`
	`void init(void)`
	`{`
	`cout<<"\n Enter rollo and grade :";`
	`cin>>rollno>>grade;`
	`cout<<"\n";`
	`}`
	`~student()`
	`{`
	`cout<<"Destroying object"<<--count<<"\n";`
	`}`
	`};`
	`int student::count=0;`
	`int main()`
	`{`
	`student classes[5];`
	`for(int i=0;i<5;i++)`
	`{`
	`cout<<"\n Enter details for student"<<i+1<<"\n";`
	`classes[i].init();`
	`}`
	`return 0;`
	`}`

Ans.	First of all the program executes default constructor as constructor is invoked automatically as soon as object is created and prints following: Creating object 1 Creating object 2 Creating object 3 Creating object 4 Creating object 5

	After that it executes init() method and asks for Entering details for 5 students as following: Enter rollno and grade: 55 A Enter rollno and grade: 24 D Enter rollno and grade: 44 A Enter rollno and grade: 67 B Enter rollno and grade: 89 B After that it executes destructor in reverse order as destructors are invoked in the reverse order in which the constructor were called and print following: Destroying object 4 Destroying object 3 Destroying object 2 Destroying object 1 Destroying object 0

11.	**Consider the following code:**
	`class ci`
	`{ int l;`
	`public:`
	`ci(int j) { l=j; }`
	`ci(ci &rv) { l=rv.l; }`
	`void initialize() { l=0; }`
	`};`
	`main()`
	`{ ci original(1);`
	`ci X1(original);`
	`ci X2=original;`
	`}`
	Referring to the sample code above, what initializes the object X1?
	i. Initialize() function
	ii. The default constructor
	iii. The copy constructor
	iv. The default copy constructor
	Justify your answer.

Ans.	The default constructor initializes the object X1 as constructor is invoked as soon as the object is created.
12.	**Which of the following is used to identify the copy constructor class type X?** **(i) (X&) (ii) X(&X) (iii) X(X&) (iv) X(X)** **Justify your answer.**
Ans.	X(X&) is used to identify the copy constructor class type X as copy constructor is a constructor of the form *classname (classame &)*.
13.	**In the case of copy constructor, which of the following is true?** **(i) Used to instantiate an object from another existing object.** **(ii) To copy one object to another existing object.** **(iii) Can be a substitute for a "=" operator for class objects.** **(iv) All of the above.**
Ans.	
14.	**What do you think is the advantage of declaring the constructor and destructor functions for public member access?**

	(i) It allows the constructor and destructor access to the data members. **(ii) It means that the constructor and destructor can be called directly by code in main() functions.** **(iii) It allows the constructor access to the other member functions.** **(iv) None of the above.**
Ans.	(ii) It means that the constructor and destructor can be called directly by code in main() functions.

15.	**Answer the question (i) and (ii) after going through the following class:**
	`class WORK`
	`{ int WorkId; char WorkType;`
	`public:`
	`~WORK() //Function 1`
	`{ cout<<"Un-Allocated"<<endl; }`
	`void status() //Function 2`
	`{ cout<<WorkId<<":"<<WorkType<<endl; }`
	`WORK() //Function 3`
	`{ WorkId=10; WorkType='T'; }`
	`Work(WORK &W) //Function 4`
	`{ WorkId=W.WorkId+12;`
	`WorkType=W.WorkType+1;`
	`}`
	`};`
	(i) Which member function out of Function 1, Function 2, Function 3 and Function 4 shown in the above definition of class WORK is called automatically, when the scope of a object gets over? Is it known as Constructor OR Destructor OR Overloaded Function OR Copy Constructor?
	(ii) WORK W; //Statement 1
	WORK Y(W); //Statement 2
	Which member function out of Function 1, Function 2, Function 3 and Function 4 shown in the above definition of class WORK will be called an execution of statement written as statement 2? What is this function specifically known as out of Destructor or Copy Constructor or Default Constructor?

Ans.	(i) Destructor (ii) Copy Constructor

16.	**Identify the error(s) in the following code and correct the code, explaining every change being introduced:**
	`#include<iostream.h>`
	`class code`
	`{ int no;`
	`char branch;`
	`static int count;`
	`code (it i=0,char b);`
	`public:`
	`code(code A)`
	`{ no=A.no;`
	`branch=A.branch;`
	`}`
	`~code()`
	`{ cout<<"Destroying Object"<<--count<<"\n";`
	`}`
	`};`
	`code(int i,char b)`
	`{ no=i;`
	`branch=b;`
	`}`
	`int main()`
	`{ code X,Y;`
	`:`
	`return 0;`
	`}`

Ans.

```
#include<iostream.h>
class code
{     int no;
      char branch;
      static int count;
public:
code(int i=0,char b);
      code(code &A)
      {     no=A.no;
            branch=A.branch;
      }
      ~code()
      {     //count=0;
            cout<<"Destroying Object"<<--count<<"\n";
      }
};
int code::count=0;
code::code(int i,char b)
{     no=i;
      branch=b;
}
int main()
{     code X,Y;
      return 0;
}
```

Changes being introduced are as following:

i. Constructor definition should be public so that it can be accessed outside the class.
ii. There should be a use of '&' operator in copy constructor.
iii. There should be a definition of the static variable outside the class definition.
iv. There is a invalid use of ':' expression.

17. **Identify the error(s) in the following code and correct the code, explaining every change being introduced in the program:**

```
#include<iostream.h>
class Sample
{     int i;
      float j;
      void Sample(void)
      {     i=0;
            j=0.0;
      }
      init()
      {     cin>>i>>j;
      }
      display()
      {     cout<<"i="<<i<<"\n";
            cout<<"j="<<j<<"\n";
      }
      void Sample(void){}
};
Sample s1,s2;
```

Ans.

```
#include<iostream.h>
class Sample
{     int i;
      float j;
```

```
      public:
      Sample()
      {     i=0;
            j=0.0;
      }
      void init()
      {     cin>>i>>j;
      }
      void display()
      {     cout<<"i="<<i<<"\n";
            cout<<"j="<<j<<"\n";
      }
};
void main()
{     Sample s1,s2;
      s1.init();
      s1.display();
}
```

Changes being introduced are as following:

i. Constructor definition should be public so that it can be accessed outside the class.
ii. Constructor should not have return type.
iii. init() and display() method should have return type.
iv. Object should be created in main() method and methods are called with the help of that object.

18. **Answer the question (i) and (ii) after going through the following program:**

```
#include<iostream.h>
#include<string.h>
class Retail
{     char Category[20];
      char Item[20];
      int Qty;
      float Price;
      Retail()        //Fuction 1
      {
            strcpy(Category,"Cereal");
            strcpy(Item,"Rice");
            Qty=100;
            Price=25;
      }
     public:
      void Show()          //Function 2
      {  cout<<Category<<"-"<<Item<<":"<<Qty<<"@"<<Price<<endl;
      }
};
void main()
{
      Retail R;  //Statement 1
      R.Show();  //Statement 2
}
```

(i) Will Statement 1 initialize all the data members for object R with the values given in the Function 1? (Yes OR No). Justify your answer suggesting the correction(s) to be made in the above code.

(ii) What shall be the possible output when the program gets executed? (Assuming, if required – the suggested correction(s) are made in the program)

Ans. (i) No. Since the default constructor Retail() is declared inside private section, it cannot initialize the objects declared outside the class. Correction needed are:

	The constructor Retail() should be declared inside public section. (ii) Cereal-Rice:100@25

19.	**Answer the question (i) and (ii) after going through the following class:**
	`class Maths`
	`{ char Chapter[20];`
	`int Marks;`
	
	`public:`
	`Maths() //Member Function 1`
	`{ strcpy(Chapter,"Geometry");`
	`Marks=10;`
	`cout<<"Chapter Initialized";`
	`}`
	`~Maths() //Member Function 2`
	`{ cout<<"Chapter Over";`
	`}`
	`};`
	(i) Name the specific features of class show by Member Function 1 and Member Function 2 in the above example.
	(ii) How would Member Function 1 and Member Function 2 gets executed?

Ans.	(i) Function 1 is Constructor. Function 2 is Destructor. (ii) Function 1 (the Constructor) will get executed every time an object of class *Maths* gets created. Function 2 (the Destructor) will get executed every time an object of class *Maths* goes out of scope i.e., its scope gets over.

20.	**Answer the question (i) and (ii) after going through the following class:**
	`class Interview`
	`{ int Month;`
	`public:`
	`Interview(int y) { Month=y; } //Constructor 1`
	`Interview(Interview &t); //Constructor 2`
	`};`
	(i) Create an object, such that it invokes constructor 1.
	(ii) Write complete definition for Constructor 2.

Ans.	(i) Interview obj1(3); (ii) Interview(Interview &t) { Month=t.Month; }

21.	**Answer the question (i) and (ii) after going through the following class:**
	`class Exam`
	`{ int Rno,MaxMarks,MinMarks,Marks;`
	`public:`
	`Exam() //Module 1`
	`{ Rno=101; MaxMarks=100;`
	`MinMarks=40;Marks=75;`
	`}`
	`Exam(int Prno,int Pmarks) //Module 2`
	`{ Rno=Prno; MaxMarks=100;`
	`MinMarks=40;Marks=Pmarks;`
	`}`
	`~Exam() //Module 3`
	`{ cout<<"Exam Over"<<endl;`
	`}`
	`void Show() //Module 4`
	`{ cout<<Rno<<":"<<MaxMarks<<":"<<MinMarks<<endl;`
	`cout<<"[MarksGot]"<<Marks<<endl;`

```
    }
};
```
(i) As per Object Oriented Programming, which concept is illustrated by Module 1 and Module 2 together?
(ii) What is Module 3 referred as? When do you think, Module 3 will be invoked/called?

Ans. (i) Constructor overloading.
(ii) Destructor. It will be invoked when scope of an object gets over.

22. **Answer the question (i) and (ii) after going through the following program:**

```
#include<iostream.h>
#include<string.h>
class Bazar
{     char Type[20];
      char Product[20];
      int Qty;
      float Price;
      Bazar()               //Fuction 1
      {
            strcpy(Type,"Electronic");
            strcpy(Product,"Calculator");
            Qty=10;
            Price=225;
      }
     public:
      voif Disp()           //Function 2
      {  cout<<Type<<"-"<<Product<<":"<<Qty<<"@"<<Price<<endl;
      }
};
void main()
{
      Bazar B;   //Statement 1
      B.Disp();  //Statement 2
}
```

(i) Will Statement 1 initialize all the data members for object B with the values given in the Function 1? (Yes OR No). Justify your answer suggesting the correction(s) to be made in the above code.
(ii) What shall be the possible output when the program gets executed? (Assuming, if required – the suggested correction(s) are made in the program)

Ans. (i) No. Since the default constructor Bazar() is declared inside private section, it cannot initialize the objects declared outside the class. Correction needed are:
The constructor Bazar () should be declared inside public section.
(ii) Electronic-Calculator:10@225

23. **Define a class Play in C++ with the following specifications:**
private members of class Play

- **Playcode** **integer**
- **PlayTitle** **25 character**
- **Duration** **float**
- **Noofscenes** **integer**

public member function of class Play

- **A constructor function to initialise Duration as 45 and Noofscence as 5.**
- **Newplay() unction to accept values for Playcode and PlayTitle.**
- **Moreinfo() function to assign the values of Duration and Noofscenes with the help of corresponding values passed as parameters to this function.**
- **Showplay() function to display all the data member on the screen.**

Ans.
```
class Play
```

```
{     int Playcode;
      char Playtitle[25];
      float Duration;
      int Noofscenes;
      public:
      Play()
      {
            Duration=45.0;
            Noofscenes=5;
      }
      void Newplay()
      {     cout<<"enter playcode: ";
            cin>>Playcode;
            cout<<"enter playtitle: ";
            gets(Playtitle);
      }
      void Moreinfo(float d,int n)
      {
            Duration=d;Noofscenes=n;
      }
      void Showplay()
      {
            cout<<"Playcode: "<<Playcode<<endl;
            cout<<"Playtitle: "<<Playtitle<<endl;
            cout<<"Duration: "<<Duration<<endl;
            cout<<"Noofscenes: "<<Noofscenes<<endl;
      }
};
```

LONG ANSWER QUESTIONS

1. **Rewrite the following program after removing the syntactical error(s), if any. Underline each correction.**

```
#include<iostream.h>
const int Dividor 5;
void main()
{
      Number=15;
      for(int Count=1;Count=<5;Count++,Number-=3)
      if(Number%Dividor==0)
            cout<<Number/Dividor;
            cout<<endl;
      else
            cout<<Number+Dividor<<endl;
```

Ans.

```
#include<iostream.h>
const int Dividor=5;
void main()
{
      int Number=15;
      for(int Count=1;Count<=5;Count++,Number-=3)
      if(Number%Dividor==0)
      {
            cout<<Number/Dividor;
            cout<<endl;
      }
      else
            cout<<Number+Dividor<<endl;
```

	}
2.	**A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and requests for the number of copies required. If the requested copies are available, the total cost of the required copies is displayed, otherwise the message "Sorry! These many copies are not in stock" is displayed.** **Design a system using a class called stock with suitable member functions and constructors.**

Ans.	`#include<iostream.h>`
	`#include<conio.h>`
	`#include<stdio.h>`
	`#include<string.h>`
	`class stock`
	`{`
	`  char author[50];`
	`  char title[50];`
	`  char pub[50];`
	`  double price;`
	`  int numcopies;`
	`public:`
	`  stock();`
	`  int access_title(char a[]);`
	`  int access_author(char b[]);`
	`  void input();`
	`  void display();`
	`  void getdata(int);`
	`};`
	`stock::stock()`
	`{`
	`  char author[50]={"abc"};`
	`  char title[50]={"efg"};`
	`  char pub[50]={"hij"};`
	`  price=500;`
	`  numcopies=50;`
	`}`
	`int stock::access_title(char a[])`
	`{`
	`  if(strcmp(title,a))`
	`  return 0;`
	`  else return 1;`
	`}`
	`int stock::access_author(char b[])`
	`{`
	`  if(strcmp(author,b))`
	`  return 0;`
	`  else return 1;`
	`}`
	`void stock::getdata(int num)`
	`{`
	`  if(numcopies>=num)`
	`  cout<<"\nCost of "<<num<<" books is Rs. "<<(price*num);`
	`  else`
	`  cout<<"\nSorry! These many copies are not in stock!";`
	`}`
	`void stock::input()`

```
{
      cout<<"\nTitle: ";
      gets(title);
      cout<<"\nAuthor:";
      gets(author);
      cout<<"\nPublisher:";
      gets(pub);
      cout<<"\nPrices:";
      cin>>price;
      cout<<"\ncopies available:";
      cin>>numcopies;
}
void stock::display()
{
     cout<<"Title: "<<title<<endl;
     cout<<"Author: "<<author<<endl;
     cout<<"Publisher: "<<pub<<endl;
     cout<<"Prices: "<<price<<endl;
     cout<<"copies available: "<<numcopies<<endl;
}
void main()
{
   clrscr();
   stock obj[2];
   int n;
   char ttle[50];
   char auth[50];
   cout<<"Enter details of 3 books";
   for(int i=0;i<2;++i)
               obj[i].input();
   cout<<endl;
   cout<<"\n Enter title of required book\n";
   gets(ttle);
   cout<<"\n Enter author of required book\n";
   gets(auth);
   for(i=0;i<2;i++)
   {
       if((obj[i].access_title(ttle))&&(obj[i].access_author(auth)))
        {
            obj[i].display();
               cout<<"\nHow many copies? ";
               cin>>n;
               obj[i].getdata(n);
        }
        else
               cout<<"\nBook unavailable";
   }
   getch();
}
```

3. **Write a program to print the score board of a cricket match in real time. The display should contain the batsman's name, runs scored, indication if out, mode by which out, bowler's score (overs played, maiden overs, runs given, wickets taken). As and when a ball is thrown, the score should be updated.**
(Hint: *Use separate arrays to store batsmen's and bowlers' information*)

Ans.

```
#include<iostream.h>
#include<conio.h>
```

```
#include<stdio.h>
class batsman
{
      public:
      char name[20];
      int run_score,indi_out;
      char modeout[20];
      void getdata()
      {
            cout<<".....Enter information for batsman......."<<endl;
            cout<<"Enter batsman's name: ";
            gets(name);
            cout<<"Enter runs scored: ";
            cin>>run_score;
            cout<<"Enter 0 if NOT-OUT or 1 if OUT: ";
            cin>>indi_out;
            if(indi_out==1)
            {
                  cout<<"Enter mode by which out: ";
                  gets(modeout);
            }
      }
      void putdata()
      {
            cout<<"......Batsman's Information........."<<endl;
            cout<<"Batsman's name: "<<name<<endl;
            cout<<"Runs scored: "<<run_score<<endl;
            if(indi_out==1)
            {
                  cout<<"OUT: "<<"Yes"<<endl;
                  cout<<"Mode by which out: "<<modeout<<endl;
            }
      }
      void update()
      {
            int new_run;
            cout<<".....Enter update for batsman....."<<endl;
            cout<<"Enter new run: ";
            cin>>new_run;
            putdata();
            run_score=run_score+new_run;
            cout<<"Updated run: "<<run_score;
      }
};
class bowler
{
      public:
      char bname[20];
      int over_play,maiden_over,run_given,wicket;
      void getinfo()
      {
            cout<<".....Enter information for bolwer......."<<endl;
            cout<<"Enter bowler's name: ";
            gets(bname);
            cout<<"Enter overs played: ";
```

```
            cin>>over_play;
            cout<<"Enter maiden overs: ";
            cin>>maiden_over;
            cout<<"Enter runs given: ";
            cin>>run_given;
            cout<<"Enter wicket taken: ";
            cin>>wicket;
      }
      void disp_info()
      {
            cout<<"......Bowler's Information........."<<endl;
            cout<<"Bolwer's name: "<<bname<<endl;
            cout<<"Overs played: "<<over_play<<endl;
            cout<<"Maiden overs: "<<maiden_over<<endl;
            cout<<"Runs given: "<<run_given<<endl;
            cout<<"Wicket taken: "<<wicket<<endl;
      }
      void upd()
      {
            int new_over,new_maidover,new_run,new_wicket;
            cout<<endl<<".....Enter update for bolwer....."<<endl;
            cout<<"Enter new overs played: ";
            cin>>new_over;
            cout<<"Enter new maiden overs: ";
            cin>>new_maidover;
            cout<<"Enter new runs given: ";
            cin>>new_run;
            cout<<"Enter new wickets taken: ";
            cin>>new_wicket;
            disp_info();
            over_play=over_play+new_over;
            maiden_over=maiden_over+new_maidover;
            run_given=run_given+new_run;
            wicket=wicket+new_wicket;
            cout<<"After update......."<<endl;
            cout<<"Overs played: "<<over_play<<endl;
            cout<<"Maiden overs: "<<maiden_over<<endl;
            cout<<"Runs given: "<<run_given<<endl;
            cout<<"Wicket taken: "<<wicket<<endl;
      }
};
void main()
{
      clrscr();
      int ch;
      batsman b1;
      bowler b2;
      b1.getdata();
      b2.getinfo();
      b1.putdata();
      b2.disp_info();
      cout<<"Is ball thrown..?? (1-Yes or 0-NO) ";
      cin>>ch;
      if(ch==1)
      {
```

```
            b1.update();
            b2.upd();
      }
      clrscr();
}
```

4. **Write a program to prepare the invoice from the following data:**
Customer number, customer name, customer address, date of sale, item no, item description, quantity sold, unit price of item, discount percentage, sales tax percentage.
Note: ***Identify different classes possible here and make sure that the date of sale becomes equal to today's date as soon as object is created. Today's date should be accepted from user, over in beginning.***

Ans.

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
class date
{
      public:
      int d,m,y;
      void getdate()
      {
            cout<<"Enter day,month and year:";
            cin>>d;cin>>m;cin>>y;
      }
      void putdate()
      {
            cout<<"Date of sale: "<<d<<"/"<<m<<"/"<<y<<"/"<<endl;
      }
};
class invo
{
      public:
      int cno;
      char cname[30];
      char add[50];
      int ino,qty;
      char desc[30];
      float price,disc_per,tax_per;
      void getinfo()
      {
            cout<<"Enter customer number: ";
            cin>>cno;
            cout<<"Enter customer Name: ";
            gets(cname);
            cout<<"Enter customer Address: ";
            gets(add);
            cout<<"Enter item number: ";
            cin>>ino;
            cout<<"Enter Quantity: ";
            cin>>qty;
            cout<<"Enter dscription: ";
            gets(desc);
            cout<<"Enter price: ";
            cin>>price;
            cout<<"Enter discount percentage: ";
            cin>>disc_per;
            cout<<"Enter tax percentage: ";
```

```
            cin>>tax_per;
      }
      void disp_info()
      {
            float disc,final,tax,tot;
            cout<<"Customer number: "<<cno<<endl;
            cout<<"Customer name: "<<cname<<endl;
            cout<<"Customer address: "<<add<<endl;
            cout<<"Item number: "<<ino<<endl;
            cout<<"Quantity: "<<qty<<endl;
            cout<<"Dscription: "<<desc<<endl;
            cout<<"Price per unit: "<<price<<endl;
            cout<<"Discount percentage: "<<disc_per<<endl;
            cout<<"Tax percentage: "<<tax_per<<endl;
            tot=price*qty;
            cout<<"Total price:"<<tot<<endl;
            disc=((tot*disc_per)/100);
            tax=((tot*tax_per)/100);
            final=(tot+tax)-disc;
            cout<<"Final Price: "<<final<<endl;

      }
};
void main()
{
      clrscr();
      date d1;
      invo i1;
      d1.getdate();
      i1.getinfo();
      d1.putdate();
      i1.disp_info();
      getch();
}
```

5. **Define a class TravelPlan in C++ with the following descriptions:**

Private Members:

- **PlanCode** **of type long**
- **Place** **of type character array (string)**
- **Number_of_travellers** **of type integer**
- **Number_of_buses** **of type integer**

Public Members:

- **A constructor to assign initial values of PlanCode as 1001, place as "Agra", Number_of_travellers as 5, Number_of_buses as 1**
- **A functio NewPlan() which allows user to enter PlanCode, Plan ad Number_of_travellers. Also, assign the value of Number_of_buses as per the following conditions:**

Number_of_travellers	*Number_of_buses*
Less than 20	1
Equal to or more than 20 and less than 40	2
Equal to 40 or more than 40	3

- **A function ShowPlan() to display the content of all the data members on screen.**

Ans.

```
class TravelPlan
```

```
{
      long PlanCode;
      char *Place;
      int Number_of_travellers;
      int Number_of_buses;
       public:
      TravelPlan()
      {
            PlanCode=1001;
            strcpy(Place,"Agra");
            Number_of_travellers=5;
            Number_of_buses=1;
      }
      void NewPlan()
      {
            cout<<"Enter Travel code, Place and Number of travellers \n";
            cin>>PlanCode;
            gets(Place);
            cin>>Number_of_travellers;
            if(Number_of_travellers<20)
                  Number_of_buses=1;
            else if(Number_of_travellers<40)
                  Number_of_buses=2;
            else
                  Number_of_buses=3;
      }
      void ShowPlan()
      {
            cout<<"Plan Code:"<<PlanCode<<endl;
            cout<<"Place:"<<Place<<endl;
            cout<<"Number of travellers:"<<Number_of_travellers<<endl;
            cout<<"Number of buses:"<<Number_of_buses<<endl;
      }
};
```

6. **Define a class Serial in C++ with the following specifications:**

private members of class Serial

- **Serialcode integer**
- **Title 25 character**
- **Duration float**
- **Noofepisodes integer**

public member function of class Play

- **A constructor function to initialise Duration as 30 and Noofepisodes as 10.**
- **Newserial() unction to accept values for Serialcode and Title.**
- **Otherentries() function to assign the values of Duration and Noofepisodes with the help of corresponding values passed as parameters to this function.**
- **Dispdata() function to display all the data member on the screen.**

Ans.

```
class Serial
{     int Serialcode;
      char Title[20];
      float Duration;
      int Noofepisodes;
      public:
      Serial()
```

```
        {
                Duration=30.0;
                Noofepisodes =10;
        }
        void Newserial()
        {       cout<<"enter Serialcode: ";
                cin>> Serialcode;
                cout<<"enter Title: ";
                gets(Title);
        }
        void Otherentries(float d,int n)
        {
                Duration=d; Noofepisodes =n;
        }
        void Dispdata()
        {
                cout<<" Serialcode: "<< Serialcode <<endl;
                cout<<" Title: "<< Title <<endl;
                cout<<"Duration: "<<Duration<<endl;
                cout<<" No of episodes: "<< Noofepisodes <<endl;
        }
};
```

7. **Define a class Clothing in C++ with the following descriptions:**

Private Members:

- **Code** **of type string**
- **Type** **of type string**
- **Size** **of type integer**
- **Material** **of type string**
- **Price** **of type float**

A function Calc_Price() which calculates and assign the values of GPrice as follows:

For the value of Material as "COTTON":

Type	Price(Rs)
TROUSER	1500
SHIRT	1200

For Material other than "COTTON" the above mentioned Price gets reduced by 25%.

Public Members:

A constructor to assign initial values of Code, Type and Material with word "NOT ASSIGNED" and Price with 0.

A function Enter() to input the values of the data members Code, Type, Size and Material and invoke the Calc_Price() function.

A function Show() to display the content of all the data members for a Clothing.

Ans.

```
class Clothing
{
        char Code[15];
        char Type[15];
        int Size;
        char Meterial[15];
        float Price;
        void Cal_Price()
        {
                if(strcmp(Material,"COTTON")==0)
                {
                        if(strcmp(Type,"TROUSER")==0)
                                Price=1500;
                        else if(strcmp(Type,"SHIRT")==0)
```

```
                        Price=1200;
            }
            else
            {     if(strcmp(Type,"TROUSER")==0)
                        Price=1500-1500*0.25;
                  else if(strcmp(Type,"SHIRT")==0)
                        Price=1200-1200*0.25;
            }
      }
      public:
      Clothing()
      {
            strcpy(Code,"NOT ASSIGNED");
            strcpy(Type,"NOT ASSIGNED");
            strcpy(Material,"NOT ASSIGNED");
            Size=0;
            Price=0;
      }
      void Enter()
      {
            cout<"Enter code";
            gets(Code);
            cout<<"\nEnter type:";
            gets(Type);
            cout<<"\nEnter Size:";
            cin>>Size;
            cout<<"\nEnter Material";
            gets(Material);
            cout<<"\nEnter Price:";
            cin>>Price;
            Calc_Price();
      }
      void Show()
      {
            cout<<"\nCode:"<<Code<<endl;
            cout<<"\nType:"<<Type<<endl;
            cout<<"\nSize:"<<Size<<endl;
            cout<<"\nMaterial:"<<Material<<endl;
            cout<<"\nPrice:"<<Price<<endl;
      }
};
```

8. **Define a class Tour C++ with the description given below:**

Private Members:

TCode	**of type string**
NoofAdults	**of type integer**
NoofKids	**of type integer**
Kilometres	**of type integer**
TotalFare	**of type float**

Public Members:

- **A constructor to assign initial values as follows:**
  - **TCode with the word "NULL"**
  - **NoofAdults as 0**
  - **NoofKids as 0**
  - **Kilometres as 0**

TotalFare as 0

- **A function AssignFare() which calculates and assign the value of the date member TotalFare as follows:**
  For each Adult

Fare(Rs)	For Kilometres
500	>=1000
300	<1000 & >=500
200	<500

For each Kid the above Fare will be 50% of the Fare mentioned in the above table.
For example:
If Distance is 850, NoofAdults=2 and NoofKids =3
Then TotalFare should be calculated as
NoofAdults*30 + NoofKids *150
i.e., 2*300+3*150=1050

- **A function EnterTour() to input the values of the data members TCode, Noofadults, NoofKids and Kilometres; and invoke the AssignFare() function**
- **A Function ShowTour() which display the content of all the data members for a Tour.**

Ans.

```
class Tour
{
     char TCode[5];
     int NoofAdults;
     int NoofKids;
     int Kilometres;
     float TotalFare;
     public:
     Tour ()
     {
          strcpy(TCode,"NULL");
          NoofAduts=0;
          NoofKids =0;
          Kilometres =0;
          TotalFare=0;
     }
     void AssignFare()
     {
          int I,j;
          TotalFare=0;
          for(i=0;i<NoofAdults;i++)
          {
               if(Kilometeres>=1000)
                      TotalFare+=500;
               else if(Kilometeres>=500)
                      TotalFare+=300;
               else
                      TotalFare+=200;
          }
          for(j=0;j<NoofKids;j++)
          {
               if(Kilometeres>=1000)
                      TotalFare+=500/2;
               else if(Kilometeres>=500)
                      TotalFare+=300/2;
               else
                      TotalFare+=200/2;
          }
```

```
    }
    void EnterTour()
    {
        cout<<"Enter value of travel code:";
        cin>>TCode;
        cout<<"Enter No. of Adults:";
        cin>>NoofAdults;
        cout<<"Enter No. of Children:";
        cin>> NoofKids;
        cout<<"Enter Distance:";
        cin>> Kilometeres;
        AssignFare();
    }
    void ShowTour()
    {
        cout<<"Travel code:"<<TCode<<endl;
        cout<<"No of Adults:"<<NoofAdults<<endl;
        cout<<"No of Children:"<< NoofKids <<endl;
        cout<<"Distance:"<< Kilometres <<endl;
        cout<<"Total Fare:"<<TotalFare<<endl;
    }
};
```

9. **Define a class Outfit in C++ with the following description:**

Private Members:

- **OCode** **of type string**
- **OType** **of type string**
- **OSize** **of type integer**
- **OFabric** **of type string**
- **OPrice** **of type float**

A function IitPrice() which calculates and assigns the value of OPrice as follows:

For the value of OFabric "DENIM",

OType	OPrice (Rs)
TROUSER	1500
JACKET	2500

For OFabric other than "DENIM" the above mentioned
OPrice gets reduced by 25%

Public Members:

A constructor to assign initial values of OCode, OType and OFabric with the word "NOT INITIALISED" and OSize and OPrice with 0.

A function Input() to input the values of the data members OCode, OType, OSize ad OFabric and invoke the InitPrice() function.

A function Display() which displays the content of all the data members for an Outfit.

Ans.

```
class Outfit
{
    char OCode[15];
    char OType[15];
    int OSize;
    char OFabric[15];
    float OPrice;
    void InitPrice()
    {
        if(strcmp(OFabric,"DENIM")==0)
        {
            if(strcmp(OType,"TROUSER")==0)
```

```
                    OPrice=1500;
               else if(strcmp(OType,"JACKET")==0)
                    OPrice=2500;
          }
          else
          {    if(strcmp(OType,"TROUSER")==0)
                    Price=1500-1500*0.25;
               else if(strcmp(OType,"JACKET")==0)
                    Price=2500-2500*0.25;
          }
     }
     public:
     Outfit()
     {
          strcpy(OCode,"NOT ASSIGNED");
          strcpy(OType,"NOT ASSIGNED");
          strcpy(OFabric,"NOT ASSIGNED");
          OSize=0;
          OPrice=0;
     }
     void Input()
     {
          cout<"Enter code";
          gets(OCode);
          cout<<"\nEnter type:";
          gets(OType);
          cout<<"\nEnter Size:";
          cin>>OSize;
          cout<<"\nEnter Material";
          gets(OFabric);
          cout<<"\nEnter Price:";
          cin>>OPrice;
          InitPrice();
     }
     void Display()
     {
          cout<<"\nCode:"<<OCode<<endl;
          cout<<"\nType:"<<OType<<endl;
          cout<<"\nSize:"<<OSize<<endl;
          cout<<"\nMaterial:"<<OFabric<<endl;
          cout<<"\nPrice:"<<OPrice<<endl;
     }
};
```