Chapter-6
Classes and Objects

Examples:

1.Specifying a class:

```
# include <iostream.h>
# include <conio.h>
class student
{
private:
 char name[30];
 int
rollno,mark1,mark2,total_marks;
protected:
 void accept()
 {
 cout<<"\n Enter data name,
rollno,
       mark1 & mark2..";
 cin>>name>>rollno>>mark1>>mark2;
}
void compute()
{
 total_marks=mark1+mark2;
}
void display()
{
 cout<<"\n Name.."<<name;
 cout<<"\n Rollno.."<<rollno;
 cout<<"\n Mark 1.."<<mark1;
 cout<<"\n Mark 2.."<<mark2;
 cout<<"\n Total
mark.."<<total_mark;
}
public:
student()

{
name[0]='\0';
rollno=mark1=mark2=total_mark=0;
cout<<"\n Constructor
executed..";
}
void execute()
{
accept();
compute();
display();
}
};
void main()
{
clrscr();
student stud;
stud.execute();
getch();}
```

Output:

```
Constructor executed..
Enter data name, rollno, mark1 & mark2..
Mani 101 75  65
Name..Mani
Rollno..101
Mark 1..75
Mark 2..65
Total mark..140
```

2.Accessing class members:

```
# include <iostream.h>
# include <conio.h>
class add
{
private:
 int a,b;
public:
 int sum;
void getdata()
{
 a=5;
 b=10;
 sum += a+b;
}
};
void main()
{
add.s;
s.getdata();
cout<<s.sum;
}
```

Output:

```
15
```

3. Member functions:

```
# include <iostream.h>
# include <conio.h>
class product
{
int code,quantity;
float price;
public:
void design_data(int c,int
q,float p)
{
 code = c;
 quantity = q;
 price = p;
}
void display()
{
cout<<"\n Code:"<<code;
cout<<"\n Quantity:"<<quantity;
```

```
cout<<"\n Price:"<<price;
}
};
void main()
{
product p;
p.assign_data(101,200,12.5);
p.display();
}
```

Output:

```
Code:101
Quantity:200
Price:12.5
```

4. Static member variables:

```
# include<iostream.h>
# include<conio.h>
class simple_static
{
int a,b,sum;
static int count;
public:
void accept()
{
cout<<"\n Enter values..";
cin>>a>>b;
sum += a+b;
count++;
}
void display()
{
cout<<"\n The sum of two
numbers.."
    <<sum;
cout<<"\n This is addition.."
    <<count;
}
};
int simple_static count=0;
void main()
{
simple_static p1,p2,p3;
p1.accept();
p1.display();
p2.accept();
p2.display();
p3.accept();
p3.display();
}
```

Output:

```
Enter values..10 20
The sum of two numbers..30
This is addition 1

Enter values..5 7
The sum of two numbers..12
This is addition 2

Enter values..9 8
The sum of two numbers..17
This is addition 3
```

Exercise:

I. Identify and correct the errors in the following:

```
# include <iostream.h>
# include <conio.h>
class x
{
public:
 int a,b;
 void init()
 {
 a=b=0;
 }
 int sum();
 int square();
};
int sum()
{
return a+b;
}
int square()
{
return sum()*sum();
}
```

Error:

Scope of the Member functions is restricted to the class specified in the function header. So the functions int sum() and int square() will not have any scope.

Correction:

The sentences int sum() and int square() should be written as

`int x::sum() and int x::square()`

II.

```
# include <iostream.h>
# include <conio.h>
clas simple
{
int num1,num2,sum=0;

protected:
accept()
{
cin>>num1>>num2;
```

```
}
public:
display()
{
sum=num1+num2;
}
};
void main()
}
simple s;
s.num1=s.num2=0;
s.accept();
s.display();
}
```

Error:

a) **int num1,num2,sum=0;**
No initialization should be made in the private member.

b) **s.num1=s.num2=0;**
Private members cannot be invoked in main function.

c) **s.accept();**
Protected members cannot be invoked in the main function.

d) **s.display();**
display() function should be called with an object.

III.

```
# include<iostream.h>
# include<conio.h>
class item
{
private:
int code,quantity;
float price;
void getdata()
{
cout<<"\n Enter
code,quantity,price";
cin>>code>>quantity>>price;
}
public:
float tax='\0';
void putdata()
{
cout<<"\n Code:"<<code;
cout<<"\n Quantity:"<<quantity;
cout<<"\n Price:"<<price;
if(quantity>100)
  tax=2500;
else
  tax=1000;
cout<<"\n Tax:"<<tax;
}
};
void main()
{item i;}
```

Complete the following based on the above program:

1.Memory allocation for instance i ***12***
2.Private data members ***code***,***quantity***, ***price***
3.Public data members ***tax***
4.Methods or data members that can be accessed by i ***putdata***

IV.

1.Define a class employee with the following specification:
private members as
empno-integer
ename-20 characters
basic-float
netpay,hra,da-float
calculate()- A function to find the answer basic+hra+da with float return type public member functions of class employee.
havedata()- A function to accept values for empno,ename,basic,hra,da and call calculate() to compute netpay.
dispdata()- A function to display all the data members on the screen.

```
# include<iostream.h>
# include<conio.h>
class employee
{
private:
int empno;
char ename[20];
float basic;
float netpay,hra,da;
public:
float calculate()
{
float netpay;
netpay=basic+hra+da;
return netpay;
}
void havedata()
{
cout<<"\n Enter empno    :";
cin>>empno;
cout<<"\n Enter emp name :";
cin>>ename;
cout<<"\n Enter basic pay:";
cin>>basic;
```

```
cout<<"\n Enter hra      :";
cin>>hra;
cout<<"\n Enter da       :";
cin>>da;
netpay=calculate();
}
void dispdata()
{
cout<<"\n Emp no   :"<<empno;
cout<<"\n Emp name:"<<empname;
cout<<"\n Basic    :"<<basic;
cout<<"\n HRA      :"<<hra;
cout<<"\n DA       :"<<da;
cout<<"\n Netpay   :"<<netpay;
}
};
void main()
{
clrscr();
employee e1;
e1.havedata;
e2.dispdata;
getch();
}
```

Output:

```
Enter empno     :3260
Enter emp name  :Jai
Enter basic pay:10000
Enter hra       :800
Enter da        :500
Emp no   :3260
Emp name:Jai
Basic    :10000
HRA      :800
DA       :500
Netpay   :11300
```

2.Define a class math with the following spacifications:
private members as
num1,num2,result- float
int()- A function to initialize num1, num2 and result to zero.
protected members as
add()- A function to add num1 and num2 and store the sum in result.
prod()- A function to multiply num1 and num2 and store the product in the result.
public members as
getdata()- A function to accept values for num1 and num2.
menu()- A function to display menu
1.Add….
2.Prod….
invoke add() when choice is 1 and invoke prod when choice is 2 and also display the result.

```
# include<iostream.h>
# include<conio.h>
class math
{
private:
float num1,num2,result;
int c;
void init()
{
num1=0.0;
num2=0.0;
result=0.0;
}
protected:
void add()
{
result=num1+num2;
cout<<"\n The sum is:"<<result;
}
void prod()
{
result=num1*num2;
cout<<"\n The product
is:"<<result;
}
public:
void getdata()
{
init();
cout<<"\n Enter numbers:";
cin>>num1>>num2;
}
void menu()
{
cout<<"\n 1.Add";
cout<<"\n 2.Multiply";
cout<<"\n Enter your choice:";
cin>>c;
if(c==1)
 add();
else
 prod();
}
};
void main()
{
clrscr();
math m1;
m1.getdata();
m1.menu();
getch();
```

```
}
```

Output:

```
Enter numbers: 10 20
1.Add
2.Multiply
Enter your choice:1
The sum is:30
```

```
Enter numbers: 10 20
1.Add
2.Multiply
Enter your choice:2
The product is:200
```