

SLIP 1

Q.1. Write the definition for a class called Cylinder that contains data members radius and height. The class has the following member functions:

- void setradius(float) to set the radius of data member
- void setheight(float) to set the height of data member
- float volume() to calculate and return the volume of the cylinder

Write a C++ program to create two cylinder objects and display each cylinder volume.

```
#include<iostream>
using namespace std;
class cylinder
{
    float radius,height;
public:
    void setRadius(float r)
    {
        radius=r;
    }
    void setHeight(float h)
    {
        height=h;
    }
    float volume()
    {
        return 3.142*radius*radius*height;
    }
};
int main()
{
    cylinder c1,c2;
    c1.setRadius(3.2);
    c1.setHeight(4.2);
    cout<<"Volume of cylinder 1 is "<<c1.volume()<<endl;
    c2.setRadius(4.6);
    c2.setHeight(5.0);
    cout<<"Volume of cylinder 1 is "<<c2.volume()<<endl;
}
```

Q.2. Write a C++ program to create a class Array that contains one float array as member.

Overload the Unary ++ and -- operators to increase or decrease the value of each element of an array. Use friend function for operator function [20 Marks]

```
#include<iostream>
using namespace std;
class Array
{
    float a[20];
    int n;
public:
    void accept()
```

```

    {
        cout<<"Enter value of n:- "<<endl;
        cin>>n;
    for(int i=0;i<n;i++)
        {
            cin>>a[i];
        }
    }
    void display()
    {
        cout<<"Display Array:- "<<endl;
        for(int i=0;i<n;i++)
        {
            cout<<"\t"<<a[i];
        }
        cout<<endl;
    }
    friend void operator++(Array a1);
    friend void operator--(Array a1);
};
void operator++(Array a1)
{
    for(int i=0;i<a1.n;i++)
    {
        a1.a[i]++;
        cout<<a1.a[i];

    }
}
void operator--(Array a1)
{
    for(int i=0;i<a1.n;i++)
    {
        a1.a[i]--;
        cout<<a1.a[i];
    }
}
int main()
{
    Array a1,a2;
    a1.accept();
    a1.display();
    cout<<"Incremented Array:- ";
    ++a1;
    cout<<endl;
    cout<<"Decrmented Array:- ";
    --a1;
}

```

SLIP 2

Q.1. Write a C++ program to create two classes Rectangle1 and Rectangle2. Compare area of both the rectangles using friend function [10 Marks]

```
#include<iostream>
using namespace std;
class rectangle1;
class rectangle
{
public:
    int l,b;
    void accept()
    {
        cout<<"Enter Length of rectangle 1:- "<<endl;
        cin>>l;
        cout<<"Enter breadth of rectangle 2:- "<<endl;
        cin>>b;
    }
    friend void cmp(rectangle,rectangle1);
};
class rectangle1
{
public:
    int l,b;

    void accept()
    {
        cout<<"Enter Length of rectangle 1:- "<<endl;
        cin>>l;
        cout<<"Enter breadth of rectangle 2:- "<<endl;
        cin>>b;
    }
    friend void cmp(rectangle,rectangle1);
};
void cmp(rectangle r1,rectangle1 r2)
{
    int area1,area2;
    area1=r1.l*r1.b;
    area2=r2.l*r2.b;
    if(area1!=area2)
    {
        cout<<"Areas of rectangles are not equal"<<endl;
    }
    else
    {
        cout<<"Areas of rectangles are equal"<<endl;
    }
}
int main()
{
```

```

    rectangle r1;
    rectangle1 r2;
    r1.accept();
    r2.accept();
    cmp(r1,r2);
}

```

Q.2. A book (ISBN) and CD (data capacity) are both types of media (id, title) objects. A person buys 10 media items, each of which can be either book or CD. Display the list of all books and CD's bought. Define the classes and appropriate member functions to accept and display data. Use pointers and concepts of polymorphism (virtual functions)

```

#include<iostream>
using namespace std;
class media
{
protected:
    int id;
    char title[20];
public:
    virtual void accept(){ }
    virtual void display(){ }
};
class book:public media
{
    int isbn;
    void accept()
    {
        cout<<"Enter Book Id:- "<<endl;
        cin>>id;
        cout<<"Enter Book Title:- "<<endl;
        cin>>title;
        cout<<"Enter ISBN:- "<<endl;
        cin>>isbn;
    }
    void display()
    {
        cout<<"Book id:- "<<id<<endl;
        cout<<"Book Title:- "<<title<<endl;
        cout<<"ISBN:- "<<isbn<<endl;
    }
};
class CD: public media
{
    int capacity;
    void accept()
    {
        cout<<"Enter CD Id:- "<<endl;
        cin>>id;
        cout<<"Enter CD Title:- "<<endl;
        cin>>title;
        cout<<"Enter Capacity:- "<<endl;
    }
};

```

```

        cin>>capacity;
    }
    void display()
    {
        cout<<"CD id:- "<<id<<endl;
        cout<<"CD Title:- "<<title<<endl;
        cout<<"Capacity:- "<<capacity<<endl;
    }
};
int main()
{
    media *m[10];
    book b[10];
    CD c[10];
    int n,n1;
    cout<<"Enter number of Books:- "<<endl;
    cin>>n;
    for(int i=0;i<n;i++)
    {
        m[i]=&b[i];
        m[i]->accept();
    }
    cout<<"Enter number of CD:- "<<endl;
    cin>>n1;
    for(int i=0;i<n1;i++)
    {
        m[i]=&c[i];
        m[i]->accept();
    }
    cout<<"Purchased Books are:- "<<endl;
    for(int i=0;i<n;i++)
    {
        m[i]=&b[i];
        m[i]->display();
    }
    cout<<"Purchased CDs are:- "<<endl;
    for(int i=0;i<n1;i++)
    {
        m[i]=&c[i];
        m[i]->display();
    }
}

```

SLIP3

Q.1. Write a C++ program to overload function volume and find volume of cube, cylinder and sphere.

[10

Marks]

Solution:-

```

#include<iostream>
using namespace std;
inline int volume(int side)
{
    return side*side;
}
inline float volume(int radius,int height)
{
    return 3.142*radius*radius*height;
}
inline double volume(double radius)
{
    return 4/3*3.142*radius*radius*radius;
}
int main()
{
    cout<<"Volume of cube:- "<<volume(5)<<endl;
    cout<<"Volume of Cylinder:- "<<volume(4,7)<<endl;
    cout<<"Volume of Sphere:- "<<volume(4.6)<<endl;
}

```

Q.2. Write a program with Student as abstract class and create derive classes Engineering, Medicine and Science from base class Student. Create the objects of the derived classes and process them and access them using array of pointer of type base class Student.

```

#include<iostream>
using namespace std;
class student
{
    public:
    virtual void accept (){}
    virtual void display(){}
};
class Engineering:public student
{
    int rno;
    char name[20];
public:
    void accept()
    {
        cout<<"Faculty: Engineering"<<endl;
        cout<<"Enter Roll No:- "<<endl;
        cin>>rno;
        cout<<"Enter Name:- "<<endl;
        cin>>name;
    }
    void display()
    {
        cout<<"Roll No:- "<<rno<<endl;
        cout<<"Name:- "<<name<<endl;
    }
};

```

```

class Medicine:public student
{
    int rno;
    char name[20];
public:
    void accept()
    {
        cout<<"Faculty: Medicine"<<endl;
        cout<<"Enter Roll No:- "<<endl;
        cin>>rno;
        cout<<"Enter Name:- "<<endl;
        cin>>name;
    }
    void display()
    {
        cout<<"Roll No:- "<<rno<<endl;
        cout<<"Name:- "<<name<<endl;
    }
};

class Science:public student
{
    int rno;
    char name[20];
public:
    void accept()
    {
        cout<<"Faculty: Science"<<endl;
        cout<<"Enter Roll No:- "<<endl;
        cin>>rno;
        cout<<"Enter Name:- "<<endl;
        cin>>name;
    }
    void display()
    {
        cout<<"Roll No:- "<<rno<<endl;
        cout<<"Name:- "<<name<<endl;
    }
};

int main()
{
    student *s;
    Engineering e;
    Medicine m;
    Science s1;
    s=&e;
    s->accept();
    s->display();
    s=&m;
    s->accept();
    s->display();
}

```

```

s=&s1;
s->accept();
s->display();
}

```

SLIP 4

Q.1. Write a C++ program to print area of circle, square and rectangle using inline function.

[10 Marks]

Solution:-

```

#include<iostream>
using namespace std;
inline int area_circle(int radius)
{
    return 3.142*radius*radius;
}
inline int area_square(int side)
{
    return side*side;
}
inline int area_rectangle(int length,int breadth)
{
    return length*breadth;
}
int main()
{
    int r,s,l,b;
    cout<<"Enter radius of circle:- "<<endl;
    cin>>r;
    cout<<"Area of circle:- "<<area_circle(r)<<endl;
    cout<<"Enter side of square:- "<<endl;
    cin>>s;
    cout<<"Area of Square:- "<<area_square(s)<<endl;
    cout<<"Enter length of rectangle:- "<<endl;
    cin>>l;
    cout<<"Enter breadth of rectangle:- "<<endl;
    cin>>b;
    cout<<"Area of rectangle:- "<<area_rectangle(l,b)<<endl;
}

```

Q.2. Create a Base class Flight containing protected data members as Flight_no, Flight_Name. Derive a class Route (Source, Destination) from class Flight. Also derive a class Reservation(Number_Of_Seats, Class, Fare, Travel_Date) from Route. Write a C++ program to perform following necessary functions:

Enter details of n reservations

Display details of all reservations

Display reservation details of a Business class

[20 Marks]

```

#include<iostream>
#include<string.h>
using namespace std;
class flight

```



```

{
    int fno;
    char fname[20];
public:
    void accept()
    {
        cout<<"Enter flight No:- "<<endl;
        cin>>fno;
        cout<<"Enter flight name:- "<<endl;
        cin>>fname;
    }
    void display()
    {
        cout<<"flight No:- "<<fno<<endl;
        cout<<"flight name:- "<<fname<<endl;
    }
};
class route:public flight
{
    char src[20],dest[20];
public:
    void accept1()
    {
        cout<<"Enter source of flight:- "<<endl;
        cin>>src;
        cout<<"Enter Destination of flight:- "<<endl;
        cin>>dest;
    }
    void display1()
    {
        cout<<"source:- "<<src<<endl;
        cout<<"Destination:- "<<dest<<endl;
    }
};
class reservation:public route
{
    int nos,fare;
    char date[20],class1[20];
public:
    void accept2()
    {
        accept();
        accept1();
        cout<<"Enter No. of seats"<<endl;
        cin>>nos;
        cout<<"Enter fare:- "<<endl;
        cin>>fare;
        cout<<"Enter class:- "<<endl;
        cin>>class1;
        cout<<"Enter date:- "<<endl;
    }
};

```

```

        cin>>date;
    }
    void display2()
    {
        display();
        display1();
        cout<<"No. of seats"<<nos<<endl;
        cout<<"fare:- "<<fare<<endl;
        cout<<"class:- "<<class1<<endl;
        cout<<"date:- "<<date<<endl;
    }
    void specific(reservation *r,int n)
    {

        for(int i=0;i<n;i++)
        {
            if(strcmp(r[i].class1,"Business")==0)
            {
                r[i].display2();
            }
        }
    }
};
int main()
{
    int n;
    reservation r[10],r1;
    cout<<"Enter no of reservations:- "<<endl;
    cin>>n;
    for(int i=0;i<n;i++)
    {
        r[i].accept2();
    }
    cout<<"Confirmed reservations are:- "<<endl;
    for(int i=0;i<n;i++)
    {
        r[i].display2();
    }
    cout<<endl;
    cout<<"Reservation of Business class are:- -"<<endl;
    r1.specific(r,n);
}

```

SLIP 5

Q.1. Write a C++ program to create a class Mobile which contains data members as Mobile_Id, Mobile_Name, Mobile_Price. Create and Initialize all values of Mobile object by using parameterized constructor. Display the values of Mobile object where Mobile_price should be right justified with a precision of two digits.

[10 Marks]

Solution:-

```
#include<iostream>
#include<string.h>
using namespace std;
class mobile
{
    int id,price;
    char name[20];
public:
    mobile(int mid,int mprice,char mname[])
    {
        id=mid;
        price=mprice;
        strcpy(name,mname);
    }
    void display()
    {
        cout<<"Display Mobile details:- "<<endl;
        cout<<"Mobile Id:- "<<id<<endl;
        cout<<"Mobile Name:- "<<name<<endl;
        cout<<"Mobile Price:- "<<price<<endl;
    }
};
int main()
{
    mobile m(1,10000,"Xiaomi");
    m.display();
}
```

Q.2. Create a class Book containing Book_name, Author and Price as a data member and write necessary member functions for the following (use function overloading).

a. To Accept and display the Book Information

b. Search book of a given author

[20 Marks]

```
#include<iostream>
#include<string.h>
using namespace std;
class Book
{
    char bname[20],author[30];
    int price;
public:
    void book(char *name,char *aname,int p)
    {
        strcpy(bname,name);
        strcpy(author,aname);
        price=p;
    }
    void book()
    {
        cout<<"Display Book Details:- "<<endl;
        cout<<"Book Name:- "<<bname<<endl;
```

```

        cout<<"Author Name:- "<<author<<endl;
        cout<<"Price:- "<<price<<endl;
    }
    void book(int n,char *aname,Book *b)
    {
        for(int i=0;i<n;i++)
        {
            if(strcmp(b[i].author,aname)==0)
            {
                b[i].book();
            }
            else
            {
                cout<<"Book not Found.."<<endl;
            }
        }
    }
};

int main()
{
    Book b[10],s;
    char name[30],aname[30],search[90];
    int price,n;
    cout<<"Enter number of Books:- "<<endl;
    cin>>n;
    cout<<"Enter the details of Books:- "<<endl;
    for(int i=0;i<n;i++)
    {
        cout<<"Enter Book Name:- "<<endl;
        cin>>name;
        cout<<"Enter Author Name:- "<<endl;
        cin>>aname;
        cout<<"Enter Price:- "<<endl;
        cin>>price;
        b[i].book(name,aname,price);
    }
    cout<<"Display all Books:- "<<endl;
    for(int i=0;i<n;i++)
    {
        b[i].book();
    }
    cout<<"Enter author name to be searched:- "<<endl;
    cin>>search;
    s.book(n,search,b);
}

```

SLIP 6

Q.1. Write a C++ program to implement a class `printdata` to overload `print` function as follows: `void print(int)` - outputs value - , that is, value followed by the value of the integer. eg. `print(10)` outputs value 10

`void print(char *)` – outputs value –“char*”, that is, value followed by the string in double quotes. eg `print(“hi”)` outputs value-“hi”

`void print(int n, char *)`- display first n characters from the given string. eg `print(3,“Object”)`- outputs value –“Obj”

```
#include<iostream>
```

```
#include<string.h>
```

```
using namespace std;
```

```
class Print{
```

```
    int x;
```

```
    char str[20];
```

```
public:
```

```
    void PRINT(int x){
```

```
        cout<<"Value:- " <<x<<endl;
```

```
    }
```

```
    void PRINT(char *str){
```

```
        cout<<"Value:- " <<str<<endl;
```

```
    }
```

```
    void PRINT(int x,char *str){
```

```
        int i;
```

```
        for(i=0;i<x;i++){
```

```
            cout<<str[i];
```

```
        }
```

```
    }
```

```
};
```

```
int main()
```

```
{
```

```
    Print p1;
```

```
    p1.PRINT(5);
```

```
    p1.PRINT("Hello");
```

```
    p1.PRINT(3,"Hello");
```

```
}
```

Q.2. Write a program to design a class `complex` to represent complex number. The `complex` class should use an external function (use it as a friend function) to add two complex number. The function should return an object of type `complex` representing the sum of two complex numbers.

```
#include<iostream>
```

```
using namespace std;
```

```
class complex
```

```
{
```

```
    int real,img;
```

```
public:
```

```
    void accept()
```

```
    {
```

```
        cout<<"Enter real number:- " <<endl;
```

```
        cin>>real;
```

```

        cout<<"Enter Imaginary number:- "<<endl;
        cin>>img;
    }
    void display()
    {
        cout<<"Display complex number:- "<<real<<"+"<<img<<"i"<<endl;
    }
    complex operator + (complex c1,complex c2){
        complex c3;
        c3.real=c1.real+c2.real;
        c3.img=c1.img+c2.img;
        cout<<"Addition of given complex number is "<<c3.real<<"+"<<c3.img<<"i"<<endl;
    }
    complex operator * (complex c1,complex c2)
    {
        complex c3;
        c3.real=c1.real*c2.real;
        c3.img=c1.img*c2.img;
        cout<<"Multiplication of given complex number is
"<<c3.real<<"+"<<c3.img<<"i"<<endl;
    }
};
int main()
{
    complex c1,c2,c3,c4;
    c1.accept();
    c1.display();
    c2.accept();
    c2.display();
    c3=c1+c2;
    c3.display();
    c4=c1*c2;
    c4.display();

}

```

SLIP 7

Q.1. Write a C++ program using class which contains two data members as type integer. Create and initialize the objects using default constructor, parameterized constructor and parameterized constructor with default value. Write a member function to display maximum from given two numbers for all objects.

```

#include<iostream>
using namespace std;
class Integer
{
    int x,y;
public:
    Integer(int a,int b)
    {
        x=a;
        y=b;
    }
}

```

```

    }
    Integer()
    {
        cout<<"Enter X:- "<<endl;
        cin>>x;
        cout<<"Enter Y:- "<<endl;
        cin>>y;
    }

```

```

void display()
{
    cout<<"X:- "<<x<<endl;
    cout<<"Y:- "<<y<<endl;

```

```

    }
};
int main()
{
    Integer i;
    i.display();
    Integer i2(20,30);
    i2.display();
}

```

Q.2. Create a class College containing data members as College_Id, College_Name, Establishment_year, University_Name. Write a program with following functions
 Accept n College details
 Display College details of specified University
 Display College details according to Establishment year (Use Array of Objects and Function Overloading)

```

#include<iostream>
using namespace std;
class college
{
    int id,eyear;
    char uname[20],cname[20];
public:
    void accept()
    {
        cout<<"Enter college Id:- "<<endl;
        cin>>id;
        cout<<"Enter college name:- "<<endl;
        cin>>cname;
        cout<<"Enter university Name:- "<<endl;
        cin>>uname;
        cout<<"Enter Establishment Year:- "<<endl;
        cin>>eyear;
    }
    void display()
    {

```

```

        cout<<"Display College Details:- "<<endl;
        cout<<"College Name:- "<<cname<<endl;
        cout<<"College Id:- "<<id<<endl;
        cout<<"University Name:- "<<uname<<endl;
        cout<<"Establishment Year:- "<<eyear<<endl;
    }
    void display(college *c,int year,int n)
    {
        for(int i=0;i<n;i++)
        {
            if(c[i].eyear==year)
            {
                c[i].display();
                break;
            }
            else
            {
                cout<<"College Not Found.."<<endl;
            }
        }
    }
};
int main()
{
    int n,year;
    college c[10],c1;
    cout<<"How many Coleges:- "<<endl;
    cin>>n;
    for(int i=0;i<n;i++)
    {
        c[i].accept();
    }
    cout<<"Display College Details:- "<<endl;
    for(int i=0;i<n;i++)
    {
        c[i].display();
    }
    cout<<"Enter year to be searched:- "<<endl;
    cin>>year;
    c1.display(c,year,n);
}

```


SLIP 8

Q.1. Write a C++ program to subtract two integer numbers of two different classes using friend function.

```
#include<iostream>
#include<string.h>
using namespace std;
class Integer1;
class Integer
{
    int x;
public:
    void accept()
    {
        cout<<"Enter Number:- "<<endl;
        cin>>x;
    }
    void display()
    {
        cout<<"X:- "<<x<<endl;
    }
    friend int sub(Integer,Integer1);
};
class Integer1
{
    int y;
public:
    void accept()
    {
        cout<<"Enter Number:- "<<endl;
        cin>>y;
    }
    void display()
    {
        cout<<"Y:- "<<endl;
    }
    friend int sub(Integer,Integer1);
};
int sub(Integer I,Integer1 I1)
{
    return I.x-I1.y;
}
int main()
{
    Integer I;
    Integer1 I1;
    I.accept();
    I.display();
    I1.accept();
    I1.display();
    cout<<"Substraction:- "<<sub(I,I1)<<endl;}
```

Q.2. Write a C++ program to create a class Date which contains three data members as dd, mm, yyyy. Create and initialize the object by using parameterized constructor and display date in dd-mon-yyyy format. (Input: 19-12-2014 Output: 19-Dec-2014) Perform validation for month. [20 Marks]

```
#include<iostream.h>
#include<conio.h>
class date
{
int dd,mm,yy;
public:
date(int d,int m,int y)
{
dd=d;
mm=m;
yy=y;
}
void display()
{
cout<<"\ngiven date is\t";
cout<<dd<<"-"<<mm<<"-"<<yy;
cout<<"\nAfter formating date is\t";
switch(mm)
{
case 1:
cout<<"\n"<<dd<<"-Jan-"<<yy;
break;
case 2:
cout<<"\n"<<dd<<"-Feb-"<<yy;
break;
case 3:
cout<<"\n"<<dd<<"-Mar-"<<yy;
break;
case 4:
cout<<"\n"<<dd<<"-Apr-"<<yy;
break;
case 5:
cout<<"\n"<<dd<<"-May-"<<yy;
break;
case 6:
cout<<"\n"<<dd<<"-Jun-"<<yy;
break;
case 7:
cout<<"\n"<<dd<<"-Jul-"<<yy;
break;
case 8:
cout<<"\n"<<dd<<"-Aug-"<<yy;
break;
case 9:
cout<<"\n"<<dd<<"-Sep-"<<yy;
break;
```

```

case 10:
cout<<"\n"<<dd<<"-Oct-"<<yy;
break;
case 11:
cout<<"\n"<<dd<<"-Nov-"<<yy;
break;
case 12:
cout<<"\n"<<dd<<"-Dec-"<<yy;
break;
default:
cout<<"\nInvalid month";
}
}
};
void main()
{
int m,dt,y;
cout<<"\n Enter date : ";
cin>>dt;
cout<<"\n Enter month : ";
cin>>m;
cout<<"\n Enter year : ";
cin>>y;
date d(dt,m,y);
d.display();
}

```

SLIP 9

Q.1. Write a C++ program to create a class Item with data members Item_code, Item_name, Item_Price. Write member functions to accept and display item information and also display number of objects created for a class.(Use Static data member and Static member function)

```

#include<iostream>
using namespace std;
class Item
{
    int code,price;
    char name[20];
    static int cnt;
public:
    void accept()
    {
        cnt++;
        cout<<"Enter Item name:- "<<endl;
        cin>>name;
        cout<<"Enter Item Code:- "<<endl;
        cin>>code;
        cout<<"Enter Item Price:- "<<endl;
        cin>>price;
    }
    void display()

```

```

{
    cout<<"Item Name:- "<<name<<endl;
    cout<<"Item Code:- "<<code<<endl;
    cout<<"Item Price:- "<<price<<endl;

}
static int count()
{

    cout<<"No of Items created:- "<<cnt<<endl;
}
};
int Item :: cnt;
int main()
{
    Item I,I2;
    I.accept();
    I.display();
    I2.accept();
    I2.display();
    I.count();
}

```

Q.2. Create a class Time which contains data members as: Hours, Minutes and Seconds. Write C++ program to perform following necessary member functions:

- i. To read time
 - ii. To display time in format like: hh:mm:ss
- To add two different times (Use Objects as argument)

```

#include <iostream.h>
class Time {
public:
    int hours;
    int minutes;
    int seconds;
    void readTime() {
        cout << "Enter time in hours, minutes and seconds: ";
        cin >> hours >> minutes >> seconds;
    }
    void displayTime() {
        cout << hours << ":" << minutes << ":" << seconds << endl;
    }
    Time addTime(Time t1, Time t2) {
        Time t3;
        t3.seconds = t1.seconds + t2.seconds;
        t3.minutes = t1.minutes + t2.minutes + (t3.seconds / 60);
        t3.hours = t1.hours + t2.hours + (t3.minutes / 60);
        t3.minutes %= 60;
        t3.seconds %= 60;
        return t3;
    }
}

```

```

};
int main() {
Time t1, t2, t3;
t1.readTime();
t2.readTime();
t3 = t3.addTime(t1, t2);
cout << "The sum of the two times is: ";
t3.displayTime();
return 0;
}

```

SLIP 10

Q.1. Write a C++ program to create a class employee containing salary as a data member. Write necessary member functions to overload the operator unary pre and post decrement “- -” for incrementing and decrementing salary.

```

#include<iostream.h>
class employee
{
int a;
public:
void accept();
void display();
void operator--();
void operator--(int);
};
void employee::accept()
{
cout<<"Enter salary : "<<endl;
cin>>a;
}
void employee::display()
{
cout<<"Salary of a : "<<a<<endl;
}
void employee::operator--()
{
a--;
cout<<"Salary of a post : "<<a<<endl;
}
void employee::operator--(int)
{
--a;
cout<<"Salary of a pre : "<<a<<endl;
}
int main()
{
employee e;
e.accept();
e.display();
e--;
--e;
}

```

```
}
```

Q.2. Design a base class Product(Product _Id, Product _Name, Price). Derive a class Discount (Discount_In_Percentage) from Product. A customer buys n Products. Calculate total price, total discount and display bill using appropriate manipulators.

[20 Marks]

```
#include<iostream.h>
#include<conio.h>
class item
{
int id;
char name[20];
public:
int price;
void accept();
void display();
int getprice();
};
class discount:public item
{
public:
int dis;
void accept1();
void display1();
};
void discount::accept1()
{
accept();
cout<<"\n enter dis";
cin>>dis;
}
void item::accept()
{
cout<<"\n enter id name and price";
cin>>id>>name>>price;
}
void item::display()
{
cout<<id<<"\t";
cout<<name<<"\t";
cout<<price<<"\t";
}
void discount::display1()
{
display();
cout<<"item dis is"<<dis<<"\n";
}
main()
{
int i,n;
```

```

discount d[10];
clrscr();
cout<<"\nenter n";
cin>>n;
for(i=0;i<n;i++)
{
d[i].accept1();
}
int dis1=0,totp=0,fp=0,totd=0;
for(i=0;i<n;i++)
{
totp=totp+d[i].price;
totd=totd+d[i].dis;
dis1=(d[i].price*d[i].dis)/100;
fp=fp+d[i].price-dis1;
}
for(i=0;i<n;i++)
{
cout<<"Item Id"<<"Item Name"<<"Item Price";
d[i].display1();
}
cout<<"\nTotal Price : "<<totp<<"\n";
cout<<"\nTotal Dis : "<<totd<<"\n";
cout<<"\n Total bill is:"<<fp<<"\n";
}

```

SLIP 11

Q.1. Write a C++ program to read two float numbers. Perform arithmetic binary operations +, -, *, / on these numbers using inline function. Display the resultant value. [10 Marks]

```

#include <iostream>
using namespace std;
inline float add(float a, float b) { return a + b; }
inline float subtract(float a, float b) { return a - b; }
inline float multiply(float a, float b) { return a * b; }
inline float divide(float a, float b) { return a / b; }
int main() {
float num1, num2;
cout << "Enter two float numbers: ";
cin >> num1 >> num2;
cout << "Addition: " << add(num1, num2) << endl;
cout << "Subtraction: " << subtract(num1, num2) << endl;
cout << "Multiplication: " << multiply(num1, num2) << endl;
cout << "Division: " << divide(num1, num2) << endl;
return 0;
}

```

Q.2. Write a C++ program to create a class Person that contains data members as Person_Name, City, Mob_No. Write a C++ program to perform following functions:

- a. To accept and display Person information
- b. To search the Person details of a given mobile number

(Use Function Overloading)

[20 Marks]

```
#include <iostream>
#include <string>
using namespace std;

class Person {
private:
    string Person_Name;
    string City;
    string Mob_No;
public:
    void acceptInfo() {
        cout << "Enter Person Name: ";
        getline(cin.ignore(), Person_Name);
        cout << "Enter City: ";
        getline(cin, City);
        cout << "Enter Mobile Number: ";
        getline(cin, Mob_No);
    }

    void displayInfo() {
        cout << "Person Name: " << Person_Name << endl;
        cout << "City: " << City << endl;
        cout << "Mobile Number: " << Mob_No << endl;
    }

    void searchPerson(const string& mobileNumber) {
        if (Mob_No == mobileNumber) {
            cout << "Person Found!" << endl;
            displayInfo();
        } else {
            cout << "Person Not Found!" << endl;
        }
    }
};

int main() {
    Person person;
    int choice;
    string mobileNumber;

    do {
        cout << "1. Accept Person Information" << endl;
        cout << "2. Display Person Information" << endl;
        cout << "3. Search Person by Mobile Number" << endl;
```



```

    cout << "4. Exit" << endl;
    cout << "Enter your choice: ";
    cin >> choice;
    cin.ignore();

    switch (choice) {
        case 1:
            person.acceptInfo();
            break;
        case 2:
            person.displayInfo();
            break;
        case 3:
            cout << "Enter Mobile Number to search: ";
            getline(cin, mobileNumber);
            person.searchPerson(mobileNumber);
            break;
        case 4:
            cout << "Exiting program..." << endl;
            break;
        default:
            cout << "Invalid choice! Please try again." << endl;
    }

    cout << endl;
} while (choice != 4);

return 0;
}

```

SLIP 12

Q.1. Write a C++ program to accept length and width of a rectangle. Calculate and display perimeter as well as area of a rectangle by using inline function.

```

#include<iostream>
using namespace std;
inline int area_rectangle(int length,int breadth)
{
    return length*breadth;
}
inline int perimeter_rectangle(int length,int breadth)
{
    return 2*(length+breadth);
}

int main()
{
    int r,s,l,b;
    cout<<"Enter length of rectangle:- " << endl;

```

```

cin>>l;
cout<<"Enter breadth of rectangle:- "<<endl;
cin>>b;
cout<<"Area of rectangle:- "<<area_rectangle(l,b)<<endl;
cout<<"Perimeter of rectangle:- "<<perimeter_rectangle(l,b)<<endl;

```

```

}

```

Q.2. Write a C++ program to implement a class 'student' to overload following functions as follows:

int maximum(int, int) – returns the maximum score of two students

int maximum(int *a, int arraylength) – returns the maximum score from an array 'a'

void maximum(int *a, int arraylength, int n) – display all students from the array 'a' having value greater than n.

```

#include <iostream>
using namespace std;
class student {
public:
int maximum(int a, int b) {
return (a > b) ? a : b;
}
int maximum(int *a, int arraylength) {
int max = a[0];
for (int i = 1; i < arraylength; i++)
if (a[i] > max)
max = a[i];
return max;
}
};
int main() {
student s;
cout << "Maximum of 2 and 3: " << s.maximum(2, 3) << endl;
int arr[] = {1, 2, 3, 4, 5};
cout << "Maximum of array: " << s.maximum(arr, 5) << endl;
return 0;
}

```

SLIP 13

Q.1. Write a C++ program to create a class which contains single dimensional integer array of given size. Define member function to display median of a given array. (Use Dynamic Constructor to allocate and Destructor to free memory of an object).

[10 Marks]

```

#include<iostream.h>
#include<conio.h>
class median
{
int *a, size;
public:
median();
void findmedian();
void display();
~median();

```

```

};
median::median()
{
int i;
cout<<"Enter size: "<<endl;
cin>>size;
a=new int[size];
for(i=0;i<size;i++)
{
cin>>a[i];
}
}
void median::findmedian()
{
if(size%2==0)
{
cout<<"Median: "<<a[(size/2)]<<endl;
}
else
{
cout<<"Median: "<<a[(size)/2]<<endl;
}
}
void median::display()
{
int i;
for(i=0;i<size;i++)
{
cout<<"\t"<<a[i];
}
cout<<endl;
}
median::~~median()
{
delete []a;
}
int main()
{
median m;
m.display();
cout<<endl;
m.findmedian();
cout<<endl;
m.display();
cout<<endl;
return 0;
}

```

Q.2. Write a C++ program to create a class Distance which contains data members as kilometer, meter. Write a C++ program to perform the following functions

To accept distance

To display distance

To overload += to add two distance

To overload > operator to compare two distance

```
#include <iostream>
using namespace std;
class Distance {
private:
    int kilometer;
    int meter;
public:
    void accept() {
        cout << "Enter kilometers: ";
        cin >> kilometer;
        cout << "Enter meters: ";
        cin >> meter;
    }
    void display() {
        cout << "Distance: " << kilometer << " km " << meter << " m" << endl;
    }
    bool operator>(const Distance &d) {
        if (kilometer > d.kilometer) {
            return true;
        } else if (kilometer == d.kilometer && meter > d.meter) {
            return true;
        } else {
            return false;
        }
    }
};

int main() {
    Distance d1, d2;
    cout << "Enter first distance:" << endl;
    d1.accept();
    cout << "Enter second distance:" << endl;
    d2.accept();
    cout << endl;
    cout << "First ";
    d1.display();
    cout << "Second ";
    d2.display();
    cout << endl;
    if (d1 > d2) {
        cout << "First distance is greater than second distance" << endl;
    } else {
        cout << "Second distance is greater than or equal to first distance" << endl;
    }
}
```

```
return 0;
}
```

SLIP 14

Q.1. Write a C++ program using class to calculate simple interest amount. (Use parameterized constructor with default value for rate)

```
#include <iostream>
using namespace std;
```

```
class SimpleInterestCalculator {
```

```
private:
```

```
    double principal;
    double rate;
    double time;
```

```
public:
```

```
    SimpleInterestCalculator(double p, double r = 0.05, double t = 1.0) {
        principal = p;
        rate = r;
        time = t;
    }
```

```
    double calculateInterest() {
        double interest = (principal * rate * time);
        return interest;
    }
```

```
};
```

```
int main() {
```

```
    double principal, rate, time;
```

```
    cout << "Enter Principal Amount: ";
    cin >> principal;
    cout << "Enter Rate of Interest (in decimal): ";
    cin >> rate;
    cout << "Enter Time Period (in years): ";
    cin >> time;
```

```
    SimpleInterestCalculator calculator(principal, rate, time);
```

```
    double interestAmount = calculator.calculateInterest();
```

```
    cout << "Principal Amount: " << principal << endl;
    cout << "Rate of Interest: " << rate << endl;
    cout << "Time Period: " << time << " years" << endl;
    cout << "Interest Amount: " << interestAmount << endl;
```

```
    return 0;
```

}

Q.2. Write a C++ program to define a class Bus with the following specifications: Bus No, Bus Name, No of Seats, Starting point, Destination .Write a menu driven program by using appropriate manipulators to

a. Accept details of n buses.

b. Display all bus details.

c. Display details of bus from specified starting point to destination

[20

Marks]

Solution:-

```
#include<iostream>
#include<string.h>
using namespace std;
class bus
{
    int bno,nos;
    char name[20],src[20],dest[20];
public:
    void accept()
    {
        cout<<endl<<"Enter Bus No:- ";
        cin>>bno;
        cout<<endl<<"Enter Bus name:- ";
        cin>>name;
        cout<<endl<<"Enter No of seats:- ";
        cin>>nos;
        cout<<endl<<"Enter source:- ";
        cin>>src;
        cout<<endl<<"Enter Destination:- ";
        cin>>dest;
    }
    void display()
    {
        cout<<endl<<"Display Bus No:- "<<bno<<endl;
        cout<<endl<<"Display Bus Name:- "<<name<<endl;
        cout<<endl<<"Display No of seats:- "<<nos<<endl;
        cout<<endl<<"Display Source:- "<<src<<endl;
        cout<<endl<<"Display Destination:- "<<dest<<endl;
    }
    void find(char *src,char *dest,bus *b,int n)
    {
        int i;
        for(i=0;i<n;i++)
        {
            if(strcmp(src,b[i].src)==0 && strcmp(dest,b[i].dest)==0)
            {
                b[i].display();
            }
        }
    }
}
```

```

    }
};
int main()
{
    bus b[10],b1;
    int n,i,ch;
    char src1[20],dest1[30];
    do
    {
        /* code */
        cout<<endl<<"1.Accept Bus details.";
        cout<<endl<<"2.Display Bus details.";
        cout<<endl<<"3.Display Bus details according to a specified starting and ending
destination."<<endl;
        cin>>ch;
        switch(ch)
        {
            case 1:cout<<endl<<"Enter no. of Buses:- "<<endl;
                cin>>n;
                for(i=0;i<n;i++)
                {
                    b[i].accept();
                }
                break;
            case 2:for(i=0;i<n;i++)
                {
                    b[i].display();
                }
                break;
            case 3:cout<<endl<<"Enter the Source:- "<<endl;
                cin>>src1;
                cout<<endl<<"Enter the Destination:- "<<endl;
                cin>>dest1;
                b1.find(src1,dest1,b,n);
                break;
            case 4:exit(0);
        }
    } while (ch!=4);
}

```

SLIP 15

Q.1. Write the definition for a class called 'point' that has x & y as integer data members. Use copy constructor to copy one object to another. (Use Default and parameterized constructor to initialize the appropriate objects). Write a C++ program to illustrate the use of above class

[10 Marks]

Solution:-

```

#include<iostream>
using namespace std;
class point
{
    int x,y;
public:
    point()
    {
        x=20;
        y=40;
        cout<<"X="<<x<<endl;
        cout<<"Y="<<y<<endl;
    }
    point(int x1,int y1)
    {
        x=x1;
        y=y1;
        cout<<"X="<<x<<endl;
        cout<<"Y="<<y<<endl;
    }
    point(point &p1)
    {
        x=p1.x;
        y=p1.y;
        cout<<"X="<<x<<endl;
        cout<<"Y="<<y<<endl;
    }
};
int main()
{
    point();
    point p2(50,60);
    point p3(p2);
}

```

Q.2. Create a base class Conversion. Derive three different classes Weight (Gram, Kilogram), Volume(Milliliter, Liter), Currency(Rupees, Paise) from Conversion class.
Write a C++ program to perform read, convert and display operations. (Use Pure virtual function) [20 Marks]

OR

Q.2. Write a C++ program to create a class Employee containing data members Emp_no, Emp_Name, Designation and Salary. Create and initialize the objects using default, parameterized and Copy Constructor. Also write member function to calculate Income tax of the employee which is 20% of salary. [20 Marks]

```

#include<iostream>

```



```

#include<string>
using namespace std;

class Employee {
private:
    int emp_no;
    string emp_name;
    string designation;
    double salary;
public:
    Employee() { // Default constructor
        emp_no = 0;
        emp_name = "";
        designation = "";
        salary = 0.0;
    }

    Employee(int en, string name, string desig, double sal) { // Parameterized constructor
        emp_no = en;
        emp_name = name;
        designation = desig;
        salary = sal;
    }

    Employee(const Employee& emp) { // Copy constructor
        emp_no = emp.emp_no;
        emp_name = emp.emp_name;
        designation = emp.designation;
        salary = emp.salary;
    }

    double calculate_income_tax() { // Member function to calculate income tax
        return salary * 0.2;
    }

    void display_details() { // Member function to display employee details
        cout << "Employee number: " << emp_no << endl;
        cout << "Employee name: " << emp_name << endl;
        cout << "Designation: " << designation << endl;
        cout << "Salary: " << salary << endl;
        cout << "Income tax: " << calculate_income_tax() << endl;
    }
};

int main() {
    Employee emp1; // Default constructor
    Employee emp2(101, "John Smith", "Manager", 50000.0); // Parameterized constructor
    Employee emp3 = emp2; // Copy constructor

    emp1.display_details();
}

```

```

    cout << endl;
    emp2.display_details();
    cout << endl;
    emp3.display_details();

    return 0;
}

```

SLIP 16

Q.1. Write a C++ program to calculate following series:

$(1) + (1+2) + (1+2+3) + (1+2+3+4) + \dots + (1+2+3+4+\dots+n)$

[10 Marks]

Solution:-

```

#include<iostream>
using namespace std;
int main()
{
    int i,n,j,sum=0;
    cout<<"Enter n:- ";
    cin>>n;
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=i;j++)
        {
            sum=sum+j;
        }
    }
    cout<<"Sum is :- " <<sum;
}

```

Q.2. Write a C++ program to read student information such as rollno, name and percentage of n students. Write the student information using file handling. [20 Marks]

```

#include <iostream>
#include <fstream>
using namespace std;
class Student {
int rno,per;
char name[20];
public :
void accept ()
{
cout<<"Student details"<<endl;
cout<<"ENTER Roll No " ;
cin>>rno;
cout<<"ENTER NAME OF THE STUDENT : ";
cin>>name;
}
}

```

```

cout<<"enter percentage";
cin>>per;
}
void displayEmployee()
{
cout<<"STUDENT ID: "<<rno<<endl;
cout<<"STUDENT NAME: "<<name<<endl;
cout<<"STUDENT PERCENTAGE:"<<per<<endl;
}
};
int main ()
{
Student s;
s.accept();
fstream file;
file.open ("stud.dat",ios:: out|ios::binary);
if (!file)
{
cout<<"Error in creating file...\n";
exit(0);
}
file.write ((char*)&s,sizeof (s));
file.close ();
cout<<"Date saved into the file.\n";
file.open ("stud.dat", ios::in|ios:: binary);
if(!file)
{
//open file again
cout<<"Error in opening file...\n";
exit (0);
}
if(file.read((char*)&s,sizeof(s)))
{
cout<<"Data from file is";
s.displayEmployee();
}
file.close();
return 0;
}

```

SLIP 17

Q.1. Write a C++ program to display factors of a number.

[10 Marks]

```

#include <iostream>
using namespace std;
int main() {
int n, i;
cout << "Enter a positive integer: ";

```

```

cin >> n;
cout << "Factors of " << n << " are: ";
for (i = 1; i <= n; ++i) {
    if (n % i == 0) {
        cout << i << " ";
    }
}
return 0;
}

```

Q.2. Design a two base classes Employee (Name, Designation) and Project(Project_Id, title). Derive a class Emp_Proj(Duration) from Employee and Project. Write a menu driven program to
 Build a master table.
 Display a master table
 Display Project details in the ascending order of duration. [20 Marks]

```

#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
#include<stdlib.h>
class employee
{
protected:
char name[20];
char dis[20];
public:
void accept()
{
cout<<"Enter name : "<<endl;
cin>>name;
cout<<"Enter designation:"<<endl;
cin>>dis;
}
void display()
{
cout<<"Name : " <<name<<endl;
cout<<"Designation : "<<dis<<endl;
}
};
class project
{
protected:
int pid;
char title[20];
public:
void accept1()
{
cout<<"Enter project id : "<<endl;
cin>>pid;
}
}

```

```

cout<<"Enter project title : "<<endl;
cin>>title;
}
void display1()
{
cout<<"Project id : "<<pid<<endl;
cout<<"Project title : "<<title<<endl;
}
};
class empproj:public employee, public project
{
protected:
int duration;
public:
void accept2()
{
accept();
accept1();
cout<<"Enter duration : "<<endl;
cin>>duration;
}
void display2()
{
display();
display1();
cout<<"Duration : "<<duration<<endl;
}
void ascending(empproj *e, int n)
{
empproj temp;
int i, j;
for(i=0;i<n;i++)
{
for(j=0;j<n-i;j++)
{
if(e[j].duration>e[j+1].duration)
{
temp=e[j];
e[j]=e[j+1];
e[j+1]=temp;
}
}
}
for(i=0;i<n;i++)
{
e[i].display2();
}
}
};
void main()

```

```

{
empproj e[10];
empproj e2;
int i, n, ch;
clrscr();
while(ch!=4)
{
cout<<"Menu "<<endl;
cout<<"1. Build master table"<<endl;
cout<<"2. Display master table"<<endl;
cout<<"3. Display project details in ascending order"<<endl;
cout<<"4. Exit"<<endl;
cout<<"Enter choice : "<<endl;
cin>>ch;
switch(ch)
{
case 1:
cout<<"Enter value of n : "<<endl;
cin>>n;
for(i=0;i<n;i++)
{
e[i].accept2();
}
break;
case 2:
cout<<"Details are : "<<endl;
for(i=0;i<n;i++)
{
e[i].display2();
}
break;
case 3:
e2.ascending(e,n);
break;
case 4:
exit(1);
default:
cout<<"Invalid chioce "<<endl;
}
}
getch();
}

```

SLIP 18

Q.1. Write a C++ program to create a class Number which contains two integer data members. Create and initialize the object by using default constructor, parameterized constructor. Write a member function to display maximum from given two numbers for all objects.

[10 Marks]

```

#include<iostream.h>
class Number
{
int x,y;
public:
Number();
Number(int,int);
void display();
};
Number::Number(void)
{
x=3;
y=5;
}
Number::Number(int a,int b)
{
x=a;
y=b;
}
void Number::display()
{
if(x>y)
cout<<&"no1 is greater than number2"&;
else
cout<<&"no2 is greater than no1"&;
}
main()
{
Number n1;
Number n2(10,20);
n1.display();
n2.display();
}

```

Q.2. Create a class Time containing members as:

- hours
- minutes
- seconds

Write a C++ program for overloading operators >> and << to accept and display a Time also write a member function to display time in total seconds. [20 Marks]

```

#include <iostream>
using namespace std;
class Time {
private:
int hours;
int minutes;
int seconds;
public:
Time() : hours(0), minutes(0), seconds(0) { }

```

```

Time(int h, int m, int s) : hours(h), minutes(m), seconds(s) {}
friend istream& operator>>(istream& in, Time& t);
friend ostream& operator<<(ostream& out, const Time& t);
void display_in_seconds() {
    cout << "Time in total seconds: " << (hours * 3600) + (minutes * 60) + seconds << endl;
}
};
istream& operator>>(istream& in, Time& t) {
    cout << "Enter hours: ";
    in >> t.hours;
    cout << "Enter minutes: ";
    in >> t.minutes;
    cout << "Enter seconds: ";
    in >> t.seconds;
    return in;
}
ostream& operator<<(ostream& out, const Time& t) {
    out << t.hours << ":" << t.minutes << ":" << t.seconds;
    return out;
}
int main() {
    Time t;
    cin >> t;
    cout << "Time entered: " << t << endl;
    t.display_in_seconds();
    return 0;
}

```

SLIP 19

Q.1. Write a C++ program to check if number is prime or not.

[10 Marks]

```

#include <iostream>
#include <math.h>
using namespace std;
int main () {
    int n, i, flag = 1;
    cout << "Enter a number: ";
    cin >> n;
    for (i = 2; i <= sqrt(n); i++) {
        if (n % i == 0) {
            flag = 0;
            break;
        }
    }
    if (n <= 1) flag = 0;
    if (flag == 1) {
        cout << n << " is a prime number";
    } else {
        cout << n << " is not a prime number"; } return 0; }

```


Q.2. Create a base class Media. Derive two different classes Book (Book_id, Book_name, Publication, Author, Book_price) and CD (CD_title, CD_price) from Media. Write a C++ program to accept and display information of both Book and CD. (Use pure virtual function)

[20 Marks]

Solution:-

```
#include<iostream>
using namespace std;
class media
{
protected:
    int id;
    char title[20];
public:
    virtual void accept(){}
    virtual void display(){}
};
class book:public media
{
    int isbn;
    void accept()
    {
        cout<<"Enter Book Id:- "<<endl;
        cin>>id;
        cout<<"Enter Book Title:- "<<endl;
        cin>>title;
        cout<<"Enter ISBN:- "<<endl;
        cin>>isbn;
    }
    void display()
    {
        cout<<"Book id:- "<<id<<endl;
        cout<<"Book Title:- "<<title<<endl;
        cout<<"ISBN:- "<<isbn<<endl;
    }
};
class CD: public media
{
    int capacity;
    void accept()
    {
        cout<<"Enter CD Id:- "<<endl;
        cin>>id;
        cout<<"Enter CD Title:- "<<endl;
        cin>>title;
        cout<<"Enter Capacity:- "<<endl;
        cin>>capacity;
    }
    void display()
    {
        cout<<"CD id:- "<<id<<endl;
```

```

        cout<<"CD Title:- "<<title<<endl;
        cout<<"Capacity:- "<<capacity<<endl;
    }
};
int main()
{
    media *m[10];
    book b[10];
    CD c[10];
    int n,n1;
    cout<<"Enter number of Books:- "<<endl;
    cin>>n;
    for(int i=0;i<n;i++)
    {
        m[i]=&b[i];
        m[i]->accept();
    }
    cout<<"Enter number of CD:- "<<endl;
    cin>>n1;
    for(int i=0;i<n1;i++)
    {
        m[i]=&c[i];
        m[i]->accept();
    }
    cout<<"Purchased Books are:- "<<endl;
    for(int i=0;i<n;i++)
    {
        m[i]=&b[i];
        m[i]->display();
    }
    cout<<"Purchased CDs are:- "<<endl;
    for(int i=0;i<n;i++)
    {
        m[i]=&c[i];
        m[i]->display();
    }
}

```

SLIP 20

Q.1. Write a C++ program to sort integer and float array elements in ascending order by using function overloading

[10

Marks]

```
#include <iostream>
```

```
#include <algorithm>
```

```
using namespace std;
```

```
void sortArray(int arr[], int n) {
```

```
sort(arr, arr + n);
```

```
}
```

```
void sortArray(float arr[], int n) {
```

```

sort(arr, arr + n);
}
int main() {
int intArr[] = {5, 3, 1, 4, 2};
float floatArr[] = {5.5, 3.3, 1.1, 4.4, 2.2};
int n = sizeof(intArr) / sizeof(intArr[0]);
sortArray(intArr, n);
sortArray(floatArr, n);
cout << "Sorted integer array: ";
for (int i = 0; i < n; i++)
cout << intArr[i] << " ";
cout << "\nSorted float array: ";
for (int i = 0; i < n; i++)
cout << floatArr[i] << " "; return 0;}

```

Q.2. Write a C++ program to create a class Department which contains data members as Dept_Id, Dept_Name, H.O.D., Number_Of_staff. Write necessary member functions to

- i. Accept details from user for 'n' departments and write it in a file "Dept.txt".
- ii. Display details of department from a file.
- iii. Count the number of objects stored in a file.

[20 Marks]

details from user for 'n' departments and write it in a file "Dept.txt". b. Display details of department from a file.

```

#include <iostream>
#include <fstream>
#include <string>
using namespace std;
class Department {
public:
int dept_id;
string dept_name;
string hod;
int num_staff;
void accept() {
cout << "Enter department ID: ";
cin >> dept_id;
cout << "Enter department name: ";
cin.ignore();
getline(cin, dept_name);
cout << "Enter H.O.D.: ";
getline(cin, hod);
cout << "Enter number of staff: ";
cin >> num_staff;
}
void write_to_file() {
ofstream file("Dept.txt", ios::app);
file << dept_id << " " << dept_name << " " << hod << " " << num_staff << endl;
file.close();
}
void display() {

```

```
cout << "Department ID: " << dept_id << ", Department Name: " << dept_name
<< ", H.O.D.: " << hod << ", Number of Staff: " << num_staff << endl;
}
};
int main() {
    int n;
    cout << "Enter number of departments: ";
    cin >> n;
    for (int i = 0; i < n; i++) {
        Department dept;
        dept.accept();
        dept.write_to_file();
    }
    ifstream file("Dept.txt");
    Department dept;
    while (file >> dept.dept_id >> dept.dept_name >> dept.hod >> dept.num_staff) {
        dept.display();
    }
}
```