**LAB 3**

**1:Write a program to create student class with data members rollno, marks1,mark2,mark3.**

**Accept data (acceptInfo()) and display using display member function.**

**Also display total,percentage and grade.**

#include<iostream>

using namespace std;

class student

{

private:

int rollno;

double m1,m2,m3;

public:

acceptinfo()

{

cout<<"Enter Roll no.- ";

cin>>rollno;

cout<<"\nEnter Marks 1 - ";

cin>>m1;

cout<<"\nEnter Marks 2 - ";

cin>>m2;

cout<<"\nEnter Marks 3 - ";

cin>>m3;

}

displayinfo()

{

double p;

cout<<"\nRoll No - "<<rollno;

cout<<"\nMarks 1 - "<<m1;

cout<<"\nMarks 2 - "<<m2;

cout<<"\nMarks 3 - "<<m3;

cout<<"\nTotal Marks - "<<m1+m2+m3;

p= ((m1+m2+m3)/300)\*100;

cout<<"\nPercentage - "<<p;

if(p>90)

cout<<"Grade = A";

else if(p>80&&p<91)

cout<<"Grade = B";

else if(p>71&&p<60)

cout<<"Grade = C";

else if(p>50&&p<61)

cout<<"Grade = D";

else

cout<<"Grade = E";

}

};

int main()

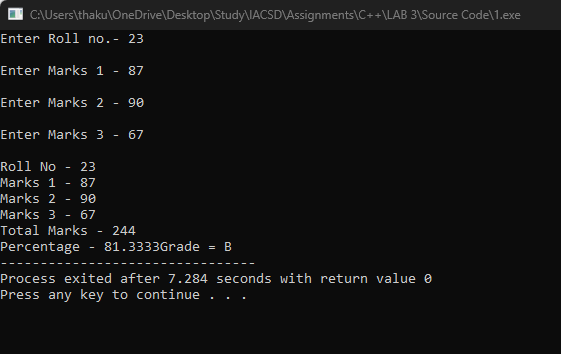
{

student s1;

s1.acceptinfo();

s1.displayinfo();

}



**2. Create a class Person with data members as name, age, city.**

**members. Also add accept and display function. . Create the**

**object of this class in main method and invoke all the methods in that class.**

#include<iostream>

using namespace std;

class person

{

private:

int age;

string name,city;

public:

acceptinfo()

{

cout<<"Enter Your Name - ";

cin>>name;

cout<<"\nEnter Your Age - ";

cin>>age;

cout<<"\nEnter Your City - ";

cin>>city;

}

displayinfo()

{

cout<<"\nName - "<<name;

cout<<"\nAge - "<<age;

cout<<"\ncity - "<<city;

}

};

int main()

{

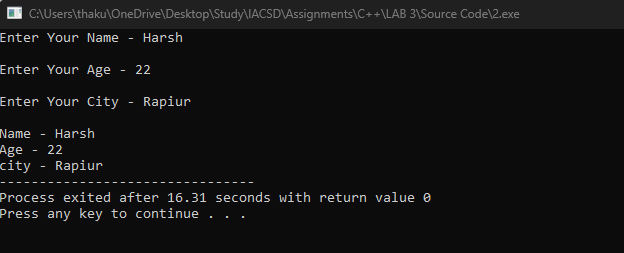
person p1;

p1.acceptinfo();

p1.displayinfo();

return 0;

}



**3. Create a class Date with data members as dd, mm, yy. Create AcceptDate function. Also add the display function. Create the**

**object of this class in main method and invoke all the methods in that class.**

#include<iostream>

using namespace std;

class date

{

private:

int dd,mm,yy;

public:

AcceptDate()

{

cout<<"Enter Date - ";

cin>>dd;

cout<<"\nEnter Month - ";

cin>>mm;

cout<<"\nEnter Year - ";

cin>>yy;

}

DisplayDate()

{

cout<<"\nDate - "<<dd<<"/"<<mm<<"/"<<yy;

}

};

int main()

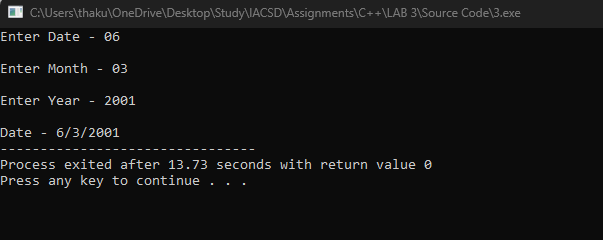
{

date d1;

d1.AcceptDate();

d1.DisplayDate();

}



**4. Create a class Book with data members as bname,id,author,price. Write AcceptBook function . Also add the display function. Create Default and Parameterized constructors. Create the object of this class in main method and invoke all the methods in that class.**

#include<iostream>

using namespace std;

class book

{

private:

int id;

string bname,author;

double price;

public:

acceptbook()

{

cout<<"Enter Book Name - ";

cin>>bname;

cout<<"\nEnter Author Name - ";

cin>>author;

cout<<"\nEnter price - ";

cin>>price;

cout<<"\nEnter id - ";

cin>>id;

}

displaybook()

{

cout<<"\nBook Name - "<<bname;

cout<<"\nAuthor Name - "<<author;

cout<<"\nEnter Price - "<<price;

cout<<"\nid - "<<id;

}

};

int main()

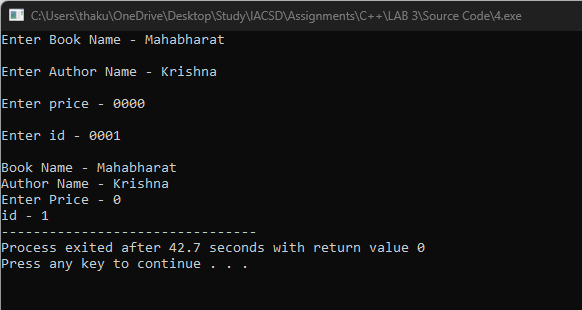
{

book b1;

b1.acceptbook();

b1.displaybook();

}



**5. Create a class Point with data members as x,y. Add AcceptPoint and add the display function. Create the object of this class in main method and invoke all the methods in that class.**

#include<iostream>

using namespace std;

class point

{

private:

int x,y;

public:

acceptpoint()

{

cout<<"Enter x cooordinate - ";

cin>>x;

cout<<"Enter y coordinate - ";

cin>>y;

}

displaypoint()

{

cout<<"Point - ("<<x<<","<<y<<")";

}

};

int main()

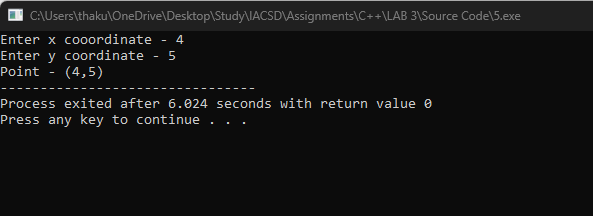
{

point p1;

p1.acceptpoint();

p1.displaypoint();

return 0;

}

**6. Create a class ComplexNumber with data members real, imaginary. Create AcceptComplexNumber() and the display function.**

**Create the object of this class in main method and invoke all the methods in that class.**

#include<iostream>

using namespace std;

class imag

{ private:

int a,b;

public:

acceptimag()

{

cout<<"Enter Real Part - ";

cin>>a;

cout<<"Enter Imaginary Part - ";

cin>>b;

}

displayimag()

{

cout<<"Imaginary Number - "<<a<<" + i"<<b;

}

};

int main()

{

imag i1;

i1.acceptimag();

i1.displayimag();

return 0;

}

