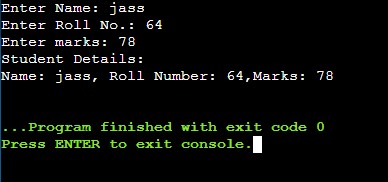
**Question 1**

#include <iostream>

using namespace std;

class stu\_details

{

private:

char name[30];

int rollno;

int marks;

public:

void get\_details(void);

void put\_details(void);

};

void stu\_details::get\_details(void){

cout<<"Enter Name: ";

cin>>name;

cout<<"Enter Roll No.: ";

cin>>rollno;

cout<<"Enter marks: ";

cin>>marks;

}

void stu\_details::put\_details(void){

cout<<"Student Details: " <<endl;

cout<<"Name: "<<name<<", Roll Number: "<<rollno<<",Marks: "<<marks<<endl;

}

int main()

{

stu\_details stud;

stud.get\_details();

stud.put\_details();

return 0;

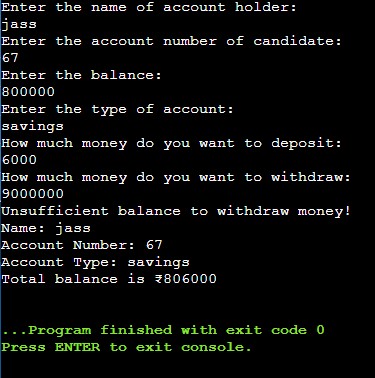
}

**Question 2**

#include <iostream>

using namespace std;

class bank{

private:

string name;

long int acc\_no;

string acc\_type;

long int balance;

public:

bank(){};

bank(string n, long int ano, string atype, int bal){

name = n;

acc\_no = ano;

acc\_type = atype;

balance = bal;

}

void set\_data(){

cout<<"Enter the name of account holder: "<<endl;

cin>>name;

cout<<"Enter the account number of candidate: "<<endl;

cin>>acc\_no;

cout<<"Enter the balance: "<<endl;

cin>>balance;

cout<<"Enter the type of account: "<<endl;

cin>>acc\_type;

}

void deposit();

void check();

void display();

};

void bank::deposit(){

long int amount;

cout<<"How much money do you want to deposit: "<<endl;

cin>>amount;

balance = balance + amount;

}

void bank::check(){

long int amt;

cout<<"How much money do you want to withdraw: "<<endl;

cin>>amt;

if(balance>amt)

balance = balance - amt;

else

cout<<"Unsufficient balance to withdraw money!"<<endl;

}

void bank::display(){

cout<<"Name: "<<name<<endl;

cout<<"Account Number: "<<acc\_no<<endl;

cout<<"Account Type: "<<acc\_type<<endl;

cout<<"Total balance is ₹"<<balance<<endl;

}

int main() {

bank customer;

customer.set\_data();

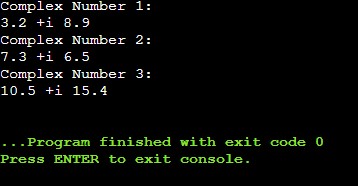
customer.deposit();

customer.check();

customer.display();

return 0;

}

**Question 3**

#include <iostream>

using namespace std;

class complex{

float real, imag;

public:

void setdata(float r, float i)

{

real = r;

imag = i;

}

void display(){

cout<<real<<" +i "<<imag<<endl;

}

complex sum(complex c){

complex n;

n.real = real + c.real;

n.imag = imag + c.imag;

return n;

}

};

int main() {

complex c1, c2, c3;

c1.setdata(3.2,8.9);

c2.setdata(7.3,6.5);

c3=c1.sum(c2);

cout<<"Complex Number 1: "<<endl;

c1.display();

cout<<"Complex Number 2: "<<endl;

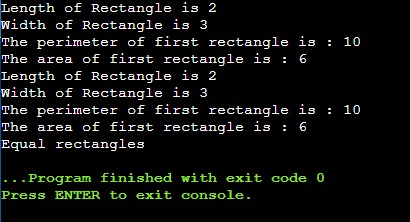
c2.display();

cout<<"Complex Number 3: "<<endl;

c3.display();

return 0;

}



**Question 4**

#include <iostream>

using namespace std;

class rectangle{

private:

float len, wid;

public:

void setlength(float l) { len = l; }

void setwidth(float w) { wid = w; }

float perimeter();

float area();

void show();

int sameArea(rectangle);

};

float rectangle::perimeter(){

float perimeter = 2\*(len + wid);

return (perimeter);

}

float rectangle::area(){

float area=len\*wid;

return (area);

}

void rectangle::show(){

cout<<"Length of Rectangle is "<<len<<endl;

cout<<"Width of Rectangle is "<<wid<<endl;

}

int rectangle::sameArea(rectangle r){

float area\_1 = len\* wid;

float area\_2 = r.len \* r.wid;

if(area\_1 == area\_2)

return 1;

else

return 0;

}

int main() {

rectangle r1, r2, r3;

r1.setlength(2);

r1.setwidth(3);

r1.show();

cout<<"The perimeter of first rectangle is : "<<r1.perimeter()<<endl;

cout<<"The area of first rectangle is : "<<r1.area()<<endl;

r2.setlength(2);

r2.setwidth(3);

r2.show();

cout<<"The perimeter of first rectangle is : "<<r2.perimeter()<<endl;

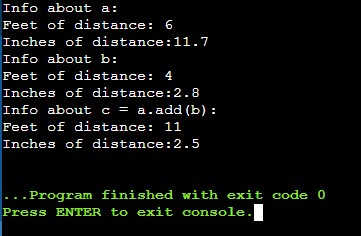
cout<<"The area of first rectangle is : "<<r2.area()<<endl;

if(r1.sameArea(r2))

cout<<"Equal rectangles";

else

cout<<"Not Equal rectangles";

}

**Question 5**

#include <iostream>

using namespace std;

class Distance

{

int feet;

float inches;

public:

Distance():feet(0),inches(0){}

void set(int ft,float in)

{

feet = ft;

if (in >= 12)

{

feet += in / 12;

int tmpin = (int)in;

inches = (tmpin % 12)+ in- tmpin;

}

else

inches = in;

}

Distance add(const Distance& d)

{

Distance x;

int tmpf = feet + d.feet;

float tmpi = inches + d.inches;

x.set(tmpf, tmpi);

return x;

}

void disp()

{

cout << "Feet of distance: " << feet

<< "\nInches of distance:" << inches<<endl;

}

};

int main()

{

Distance a;

a.set(5, 23.7);

cout << "Info about a:\n";

a.disp();

Distance b;

b.set(3, 14.8);

cout << "Info about b:\n";

b.disp();

Distance c = a.add(b);

cout << "Info about c = a.add(b):\n";

c.disp();

}