***OOP LAB ASSIGNMNET = 4***

Name :- Ankit Senjaliya

Enrollment No. :- 19BT04046

1. Write a C++ program to demonstrate an overloading of – (Binary) operator.

#include<iostream.h>

#include<conio.h>

class OVERLOADING

{

int a,b;

public:

void accept()

{

cout<<"\n\n\t\t\t Enter The First Number = ";

cin>>a;

cout<<"\n\t\t\t Enter The Second Number = ";

cin>>b;

}

OVERLOADING operator - (OVERLOADING obj)

{

OVERLOADING c;

c.a = a - obj.a;

c.b = b - obj.b;

return (c);

}

void display()

{

cout<<a<<"-"<<b<<"i"<<endl;

}

};

int main()

{

clrscr();

cout<<"\n Name :- Ankit Senjaliya ";

cout<<"\n Enrollment No. :- 19BT04046 \n\n";

OVERLOADING c1,c2,c;

c1.accept();

c2.accept();

c = c1 - c2;

cout<<"\n\n\t\t\t Entered Value = "<<endl;

cout<<"\t\t\t ";

c1.display();

cout<<"\t\t\t ";

c2.display();

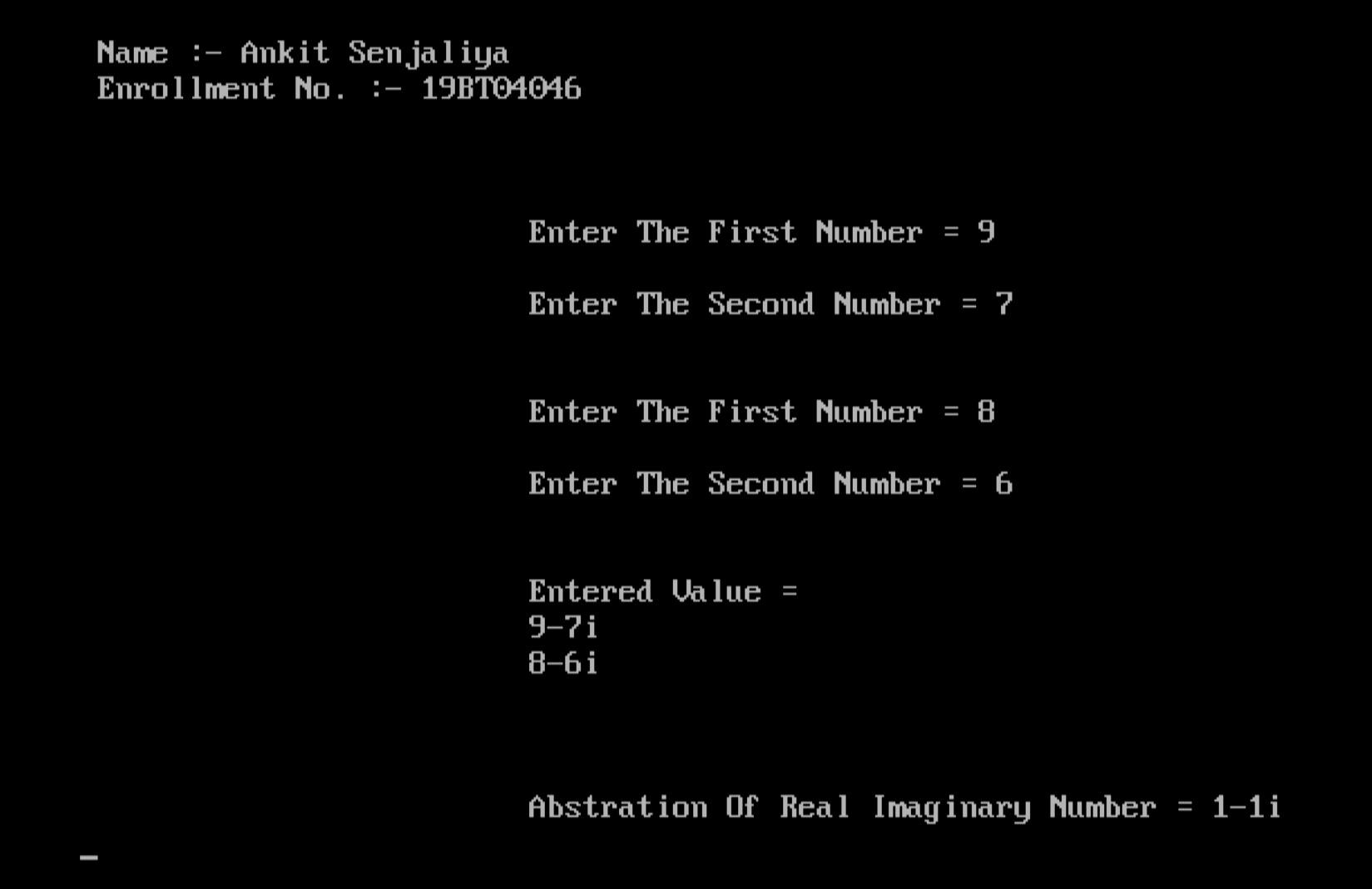
cout<<"\n\n\n\t\t\t Abstration Of Real Imaginary Number = ";

c.display();

getch();

return 0;

}



1. Write a c++ program to demonstrate an overloading of -- (unary) operator.

#include<iostream.h>

#include<conio.h>

class complex

{

int a,b;

public:

complex()

{

}

void getvalue()

{

cout<<"\n\n\t\t\t Enter The First Number = ";

cin>>a;

cout<<"\n\n\t\t\t Enter The Second Number = ";

cin>>b;

}

void operator++()

{

a = ++a;

b = ++b;

}

void operator--()

{

a = --a;

b = --b;

}

void display()

{

cout<< a <<"+"<< b <<"i"<<endl;

}

};

int main()

{

clrscr();

cout<<"\n\n\t\t\t Name :- Ankit Senjaliya ";

cout<<"\n\n\t\t\t Enrollment No. :- 19BT04046 \n\n";

complex obj;

obj.getvalue();

obj++;

cout<<"\n\n\t\t\t Increment Complex Number = ";

obj.display();

obj--;

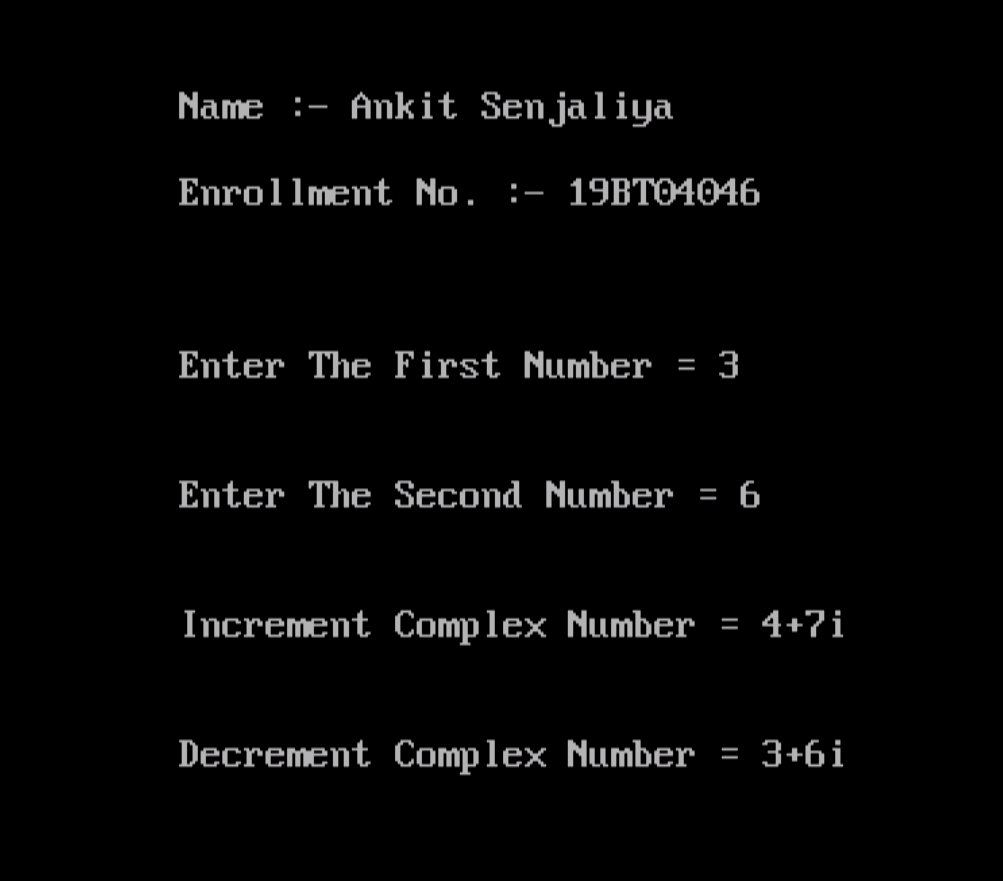
cout<<"\n\n\t\t\t Decrement Complex Number = ";

obj.display();

getch();

return 0;

}



(3) Define a class complex with real and imaginary as two data member, add necessary constructors and member function to initialize and display data of class. Class should overload the + operator to add two complex objects and return the results. Invoke the statements like C3 = C1 + C2 in main ().

#include<iostream.h>

#include<conio.h>

class complex

{

float real,imag;

public:

complex(float r, float i)

{

real = r;

imag = i;

}

void display()

{

cout<<"\n\n\t\t\t The Value Of Real = "<<real;

cout<<"\n\n\t\t\t The Value Of Imag = "<<imag;

}

friend complex operator + (complex C1, complex C2)

{

C2.real = C2.real + C1.real;

C2.imag = C2.imag + C1.imag;

return C2;

}

};

int main()

{

clrscr();

cout<<"\n\n\t\t\t Name :- Ankit Senjaliya ";

cout<<"\n\n\t\t\t Enrollment No. :- 19BT04046 \n\n";

complex C1(10,20);

complex C2(20,10);

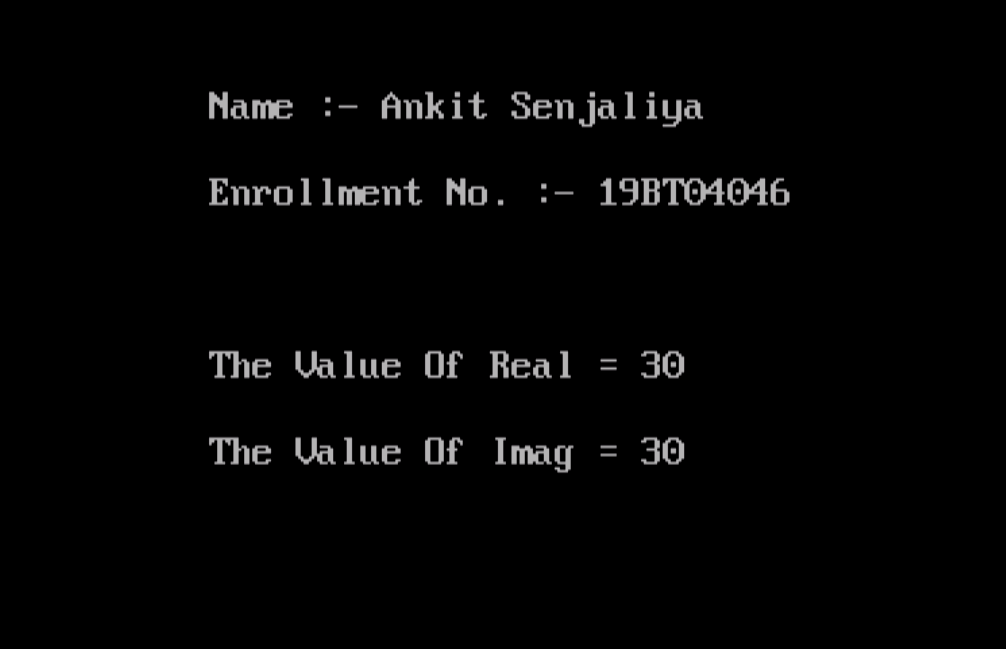
complex C3=C1+C2;

C3.display();

getch();

return 0;

}



(4) Declare a class called bird having private data members name and weight. Define following functions : - default constructor for reading data members from key board - overloaded constructor with two arguments to be used for initialization of data members. - display function to display data members. - overloaded member operator >= to compare weight of two bird objects, returning false if weight of first bird object is less than that of the second & true otherwise. Define main to illustrate use of above functions.

#include<iostream.h>

#include<conio.h>

class birds

{

char \*name,n[20];

int weight;

public:

birds();

birds(char \*,int);

void display();

friend void operator > (birds,birds);

};

birds::birds()

{

cout<<"\n\n\t\t\t Enter Of Birds Nmae = ";

cin>>name;

cout<<"\n\n\t\t\t Enter Of Birds Weight = ";

cin>>weight;

}

void birds::display()

{

cout<<"\n\n\t\t\t Name = "<<name;

cout<<"\n\n\t\t\t Weight = "<<weight<<"\n";

}

birds::birds(char \*n1, int w)

{

name = n1;

weight = w;

}

void operator > (birds a, birds b)

{

if(a.weight > b.weight)

{

cout<<"\n\n\t\t\t First Bird Weight "<<a.weight<<" Is Greater ";

}

else

{

cout<<"\n\n\t\t\t Second Bird Weight "<<b.weight<<" Is Greater ";

}

}

int main()

{

clrscr();

cout<<"\n\n\t\t\t Name :- Ankit Senjaliya ";

cout<<"\n\n\t\t\t Enrollment No. :- 19BT04046 \n";

birds a,b("Pigeon",30);

a.display();

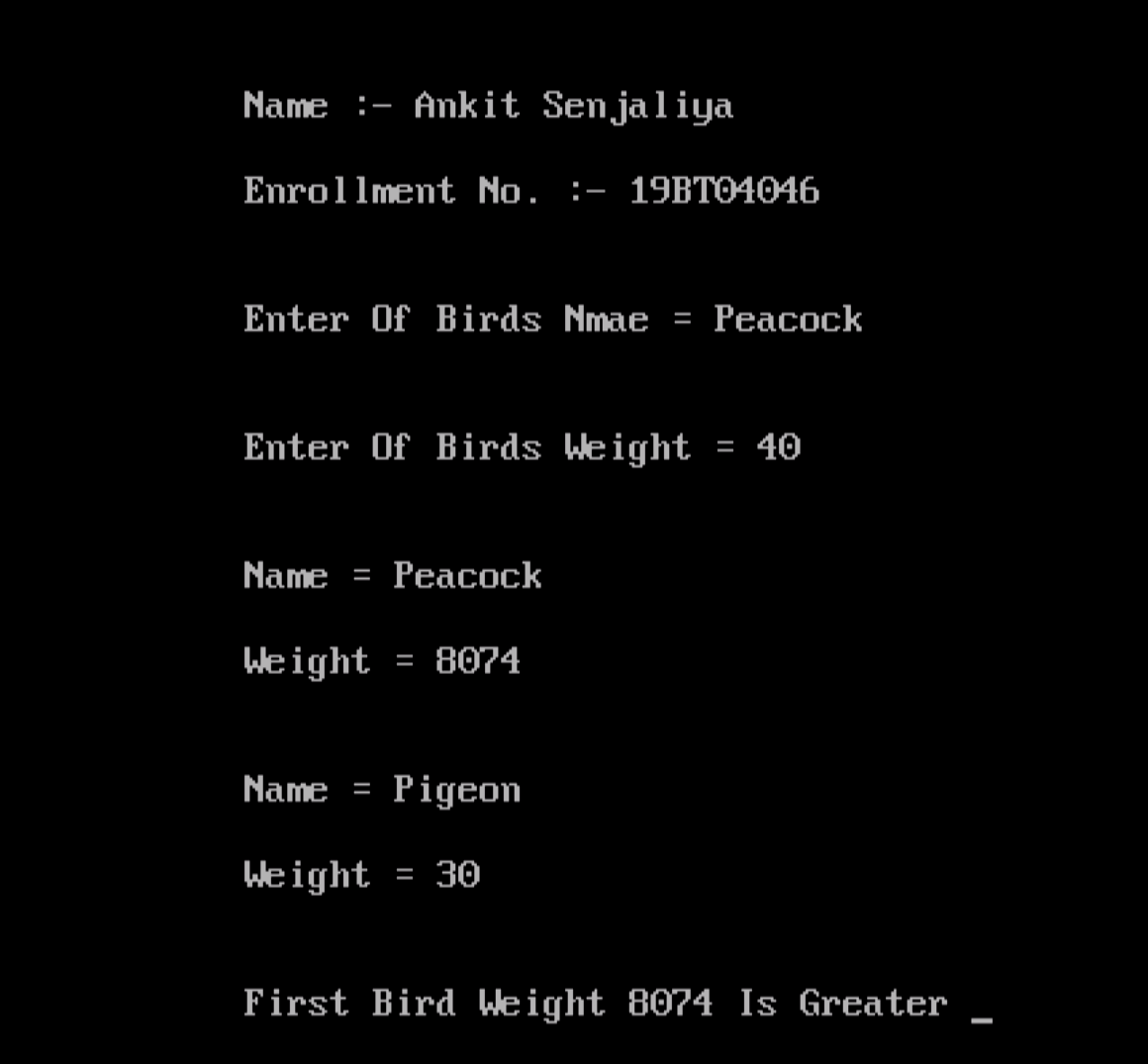
b.display();

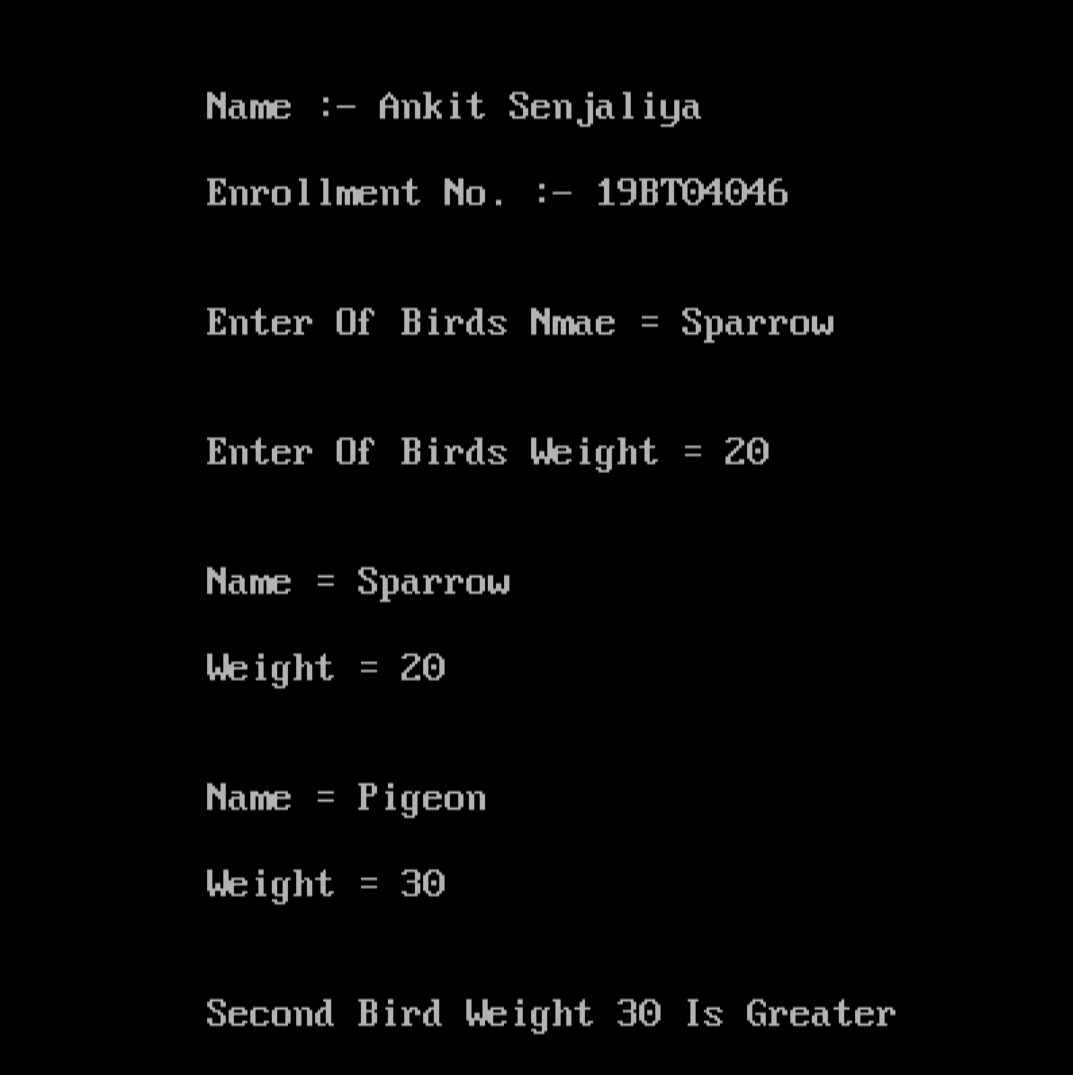
a>b;

getch();

return 0;

}





(5) Create two classes X and Y containing private variables x and y respectively. Using a common friend function, perform multiplication operation between x and y.

#include<iostream.h>

#include<conio.h>

class Y;

class X

{

private:

int numX;

public:

X(): numX(10)

{

}

friend int multiplication(X,Y);

};

class Y

{

private:

int numY;

public:

Y(): numY(20)

{

}

friend int multiplication(X,Y);

};

int multiplication(X objectX, Y objectY)

{

return (objectX.numX \* objectY.numY);

}

int main()

{

clrscr();

cout<<"\n\n\t\t\t Name :- Ankit Senjaliya ";

cout<<"\n\n\t\t\t Enrollment No. :- 19BT04046 \n\n";

X objectX;

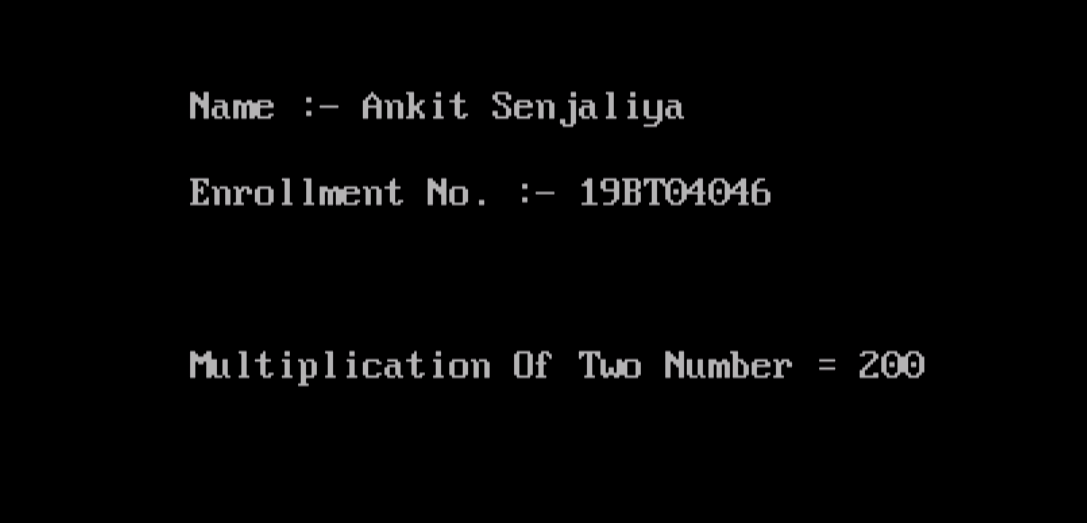
Y objectY;

cout<<"\n\n\t\t\t Multiplication Of Two Number = "<< multiplication(objectX, objectY);

getch();

return 0;

}



(6) Declare a class called book having members like book\_title, publisher and author\_name. Overload extractor and inserter operators ( >> and << ) for class book.

#include<iostream.h>

#include<conio.h>

class book

{

char book\_title[60];

char publisher[60];

char author\_name[60];

public:

friend istream & operator >> (istream &, book &);

friend ostream & operator << (ostream &, book &);

};

istream & operator >> (istream &din, book &b)

{

din>>b.book\_title;

din>>b.publisher;

din>>b.author\_name;

return(din);

}

ostream & operator << (ostream &dout, book &b)

{

dout<<b.book\_title<<"\n";

dout<<b.publisher<<"\n";

dout<<b.author\_name<<"\n";

return(dout);

}

int main()

{

clrscr();

cout<<"\n Name :- Ankit Senjaliya ";

cout<<"\n Enrollment No. :- 19BT04046 \n\n";

book m;

cout<<"\n Enter The Book Title, Publisher And Author Name = \n " ;

cin>>m;

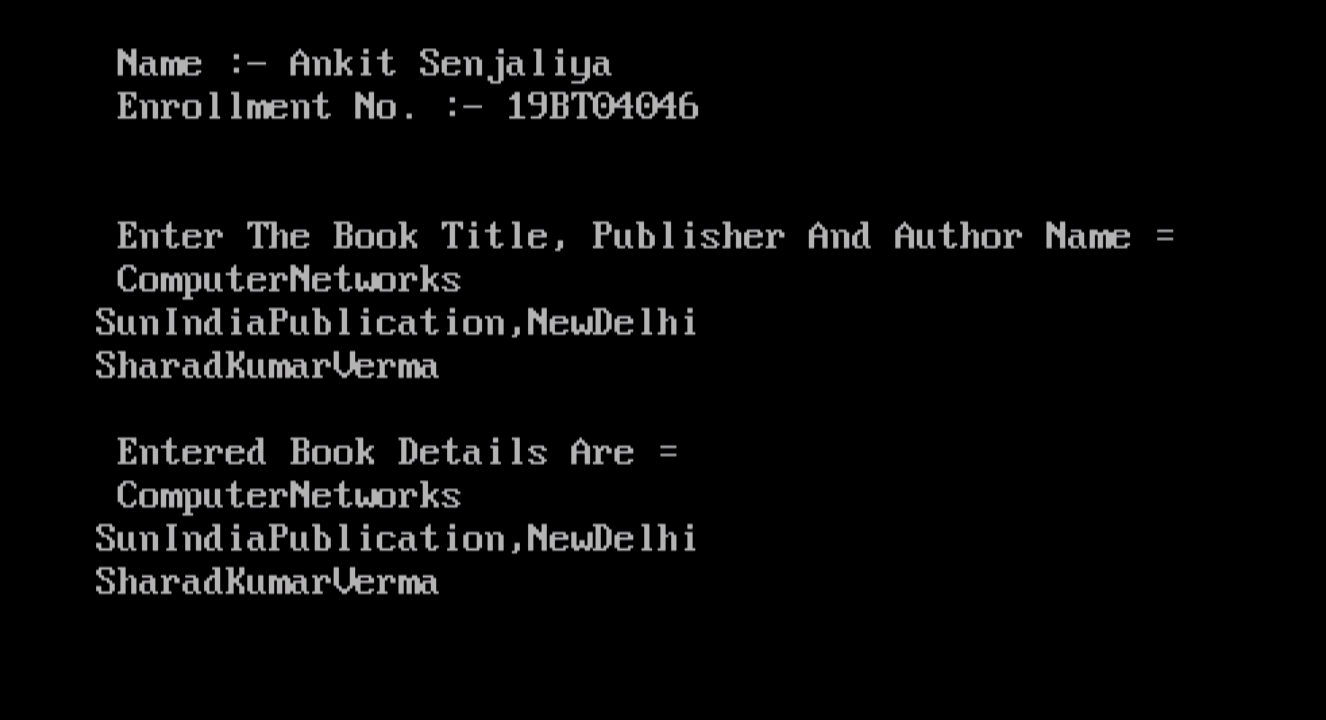
cout<<"\n Entered Book Details Are = \n ";

cout<<m;

getch();

return 0;

}



(7) overload +=,==,++ operator using friend function.

#include<iostream.h>

#include<conio.h>

//using namespace std;

class Point

{

int a,b;

public:

Point()

{}

Point(int pa, int pb)

{

a = pa;

b = pb;

}

void show()

{

cout<<"\n\n\t\t\t Value Of A = "<<a;

cout<<"\n\n\t\t\t Value Of B = "<<b;

}

friend Point operator + (Point M, Point N);

Point operator - (Point N);

Point operator = (Point N);

Point operator++();

};

Point operator + (Point M, Point N)

{

Point temp;

temp.a = M.a + N.a;

temp.b = M.b + N.b;

return temp;

}

Point Point::operator - (Point N)

{

Point temp;

temp.a = a - N.a;

temp.b = b - N.b;

return temp;

}

Point Point::operator = (Point N)

{

a = N.a;

b = N.b;

return \*this;

}

Point Point::operator++()

{

a++;

b++;

return \*this;

}

int main()

{

clrscr();

cout<<"\n\n\t\t\t Name :- Ankit Senjaliya ";

cout<<"\n\n\t\t\t Enrollment No. :- 19BT04046 \n\n";

Point K(80,70),L(70,80);

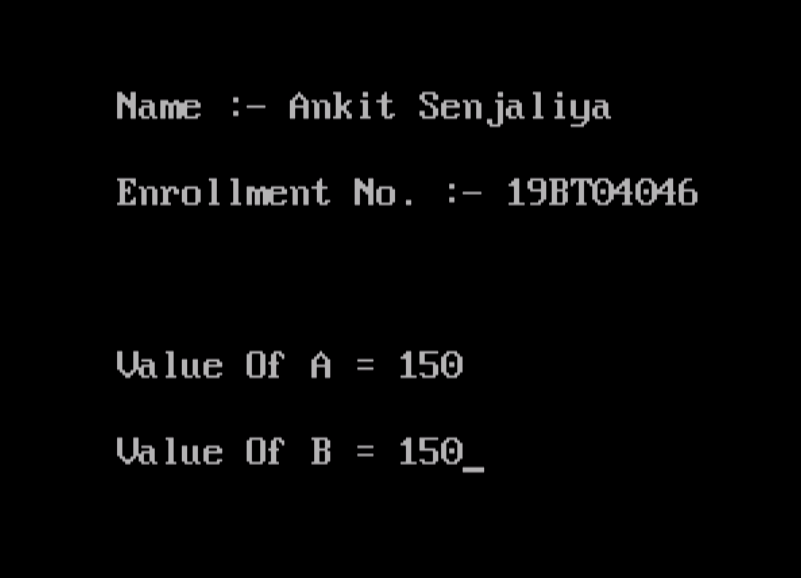
L = K + L;

L.show();

getch();

return 0;

}



(8) Write simple code to demonstrate working of static keyword with class data member and class member function (Use string as an input).

#include<iostream.h>

#include<conio.h>

class student

{

private:

char name[20];

int rollno;

int marks;

public:

static int objectCount;

student()

{

objectCount++;

}

void getdata()

{

cout<<"\n\n\t\t\t Enter Name = ";

cin>>name;

cout<<"\t\t\t Enter Roll No. = ";

cin>>rollno;

cout<<"\t\t\t Enter Marks = ";

cin>>marks;

}

void putdata()

{

cout<<"\n\t\t Name = "<<name;

cout<<"\n\t\t Roll No. = "<<rollno;

cout<<"\n\t\t Marks = "<<marks;

}

};

int student::objectCount = 0;

int main()

{

clrscr();

cout<<"\n Name :- Ankit Senjaliya ";

cout<<"\n Enrollment No. :- 19BT04046 \n\n";

student s1;

s1.getdata();

s1.putdata();

student s2;

s2.getdata();

s2.putdata();

cout<<"\n\n\t\t\t Total Objects Are Created = "<<student::objectCount;

getch();

return 0;

}

