***NAME: ROHAN GHOSH***

***ID:181001001122***

***BATCH:BCS2B***

1**. Create a class Stack, overload unary ++ (increment) andunary - - (decrement) operator in such a way ---**

**Stack ob;**

***a. ++ob; // implement PUSH function***

***b. - - ob; // implement POP function***

**There should be a display function to display the stack object**

**present status.**

**ANS: CODE**

#include<iostream>

using namespace std;

class Stack{

int top=-1,stk[5],item;

public:

void operator ++(){

if(top==4)

cout<< "\nStack is overflow...!!!\n";

else{

cout<< "\nEnter Item = ";

cin>> item;

stk[++top]=item;

}

}

void operator --(){

if(top==-1)

cout<< "\nStack is underflow...!!!\n";

else{

item=stk[top--];

cout<< "\nPopped Item = " << item;

}

}

void display(){

if(top==-1)

cout<< "\nStack is underflow...!!!\n";

else{

cout<< "\nElements in the stack are :\n";

for(int i=top;i>=0;i--)

cout<<stk[i] << "\t";

}

}

};

int main(){

int ch;

Stack ob;

cout<< "\nEnter 1 for Push"<<"\nEnter 2 for Pop."<< "\nEnter 3 for Dislay.\n";

do{

cout<< "\nEnter you choice : ";

cin>>ch;

switch(ch){

case 1:

++ob;

break;

case 2:

--ob;

break;

case 3:

ob.display();

break;

default:

cout<<"Wrong Choice";

break;

}

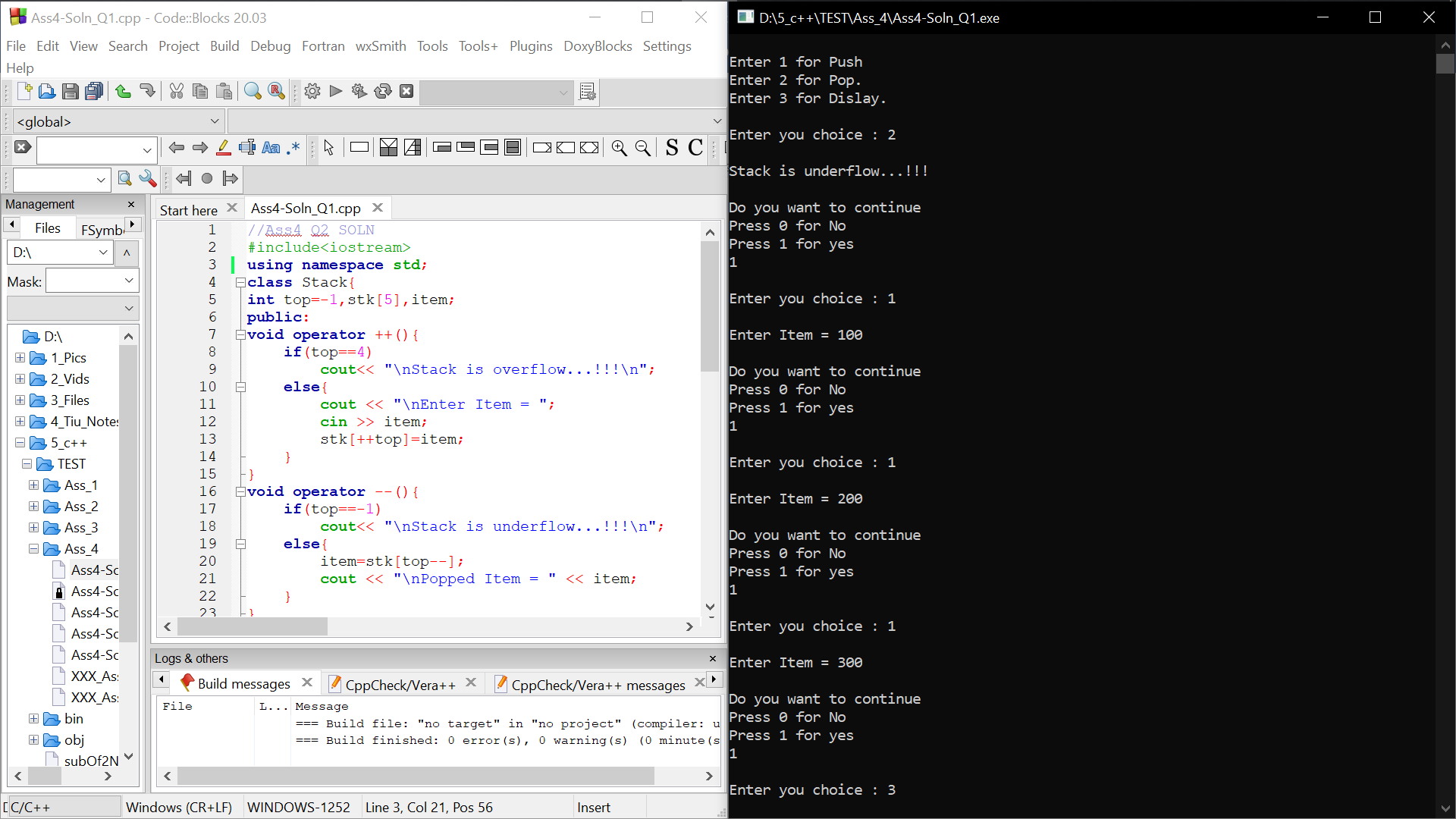
cout<< "\nDo you want to continue \nPress 0 for No\nPress 1 for yes\n";

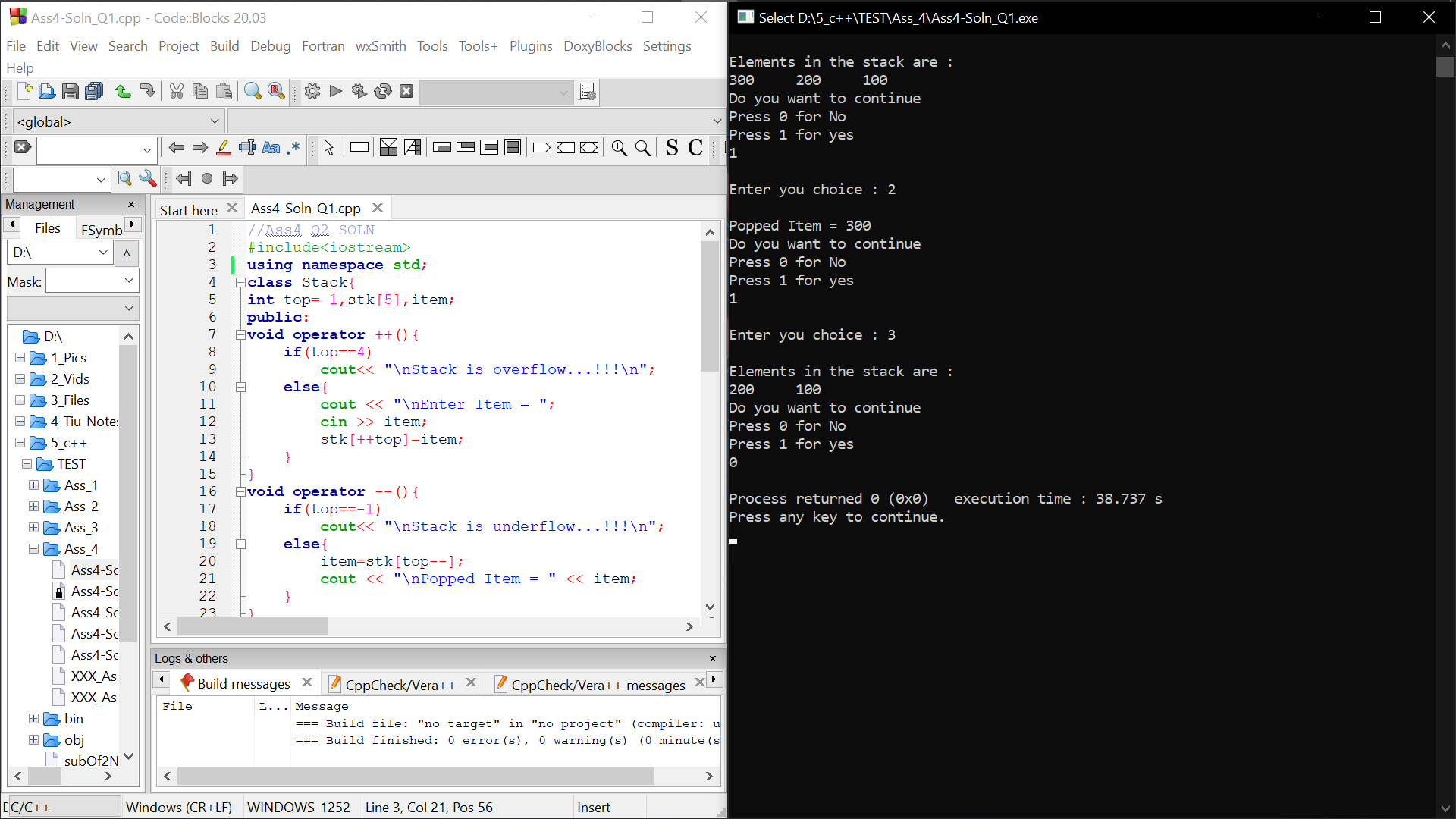
cin>>ch;

}while(ch==1);

}

**OUTPUT**





2. **Write a program to demonstrateBasic type to Distance type conversion.**

***class Distance{***

***foot, inch;***

***};***

**ANS: CODE**

#include<iostream>

using namespace std;

class Distance{

int foot,inch;

public:

Distance(){}

void operator=(int x){

foot=x/12;

inch=x%12;

}

void display(){

cout<<"\nfoot = "<<foot<<"\ninch = "<<inch;

}

};

int main(){

Distance x;

int m;

cout<<"Enter distance in inches : ";

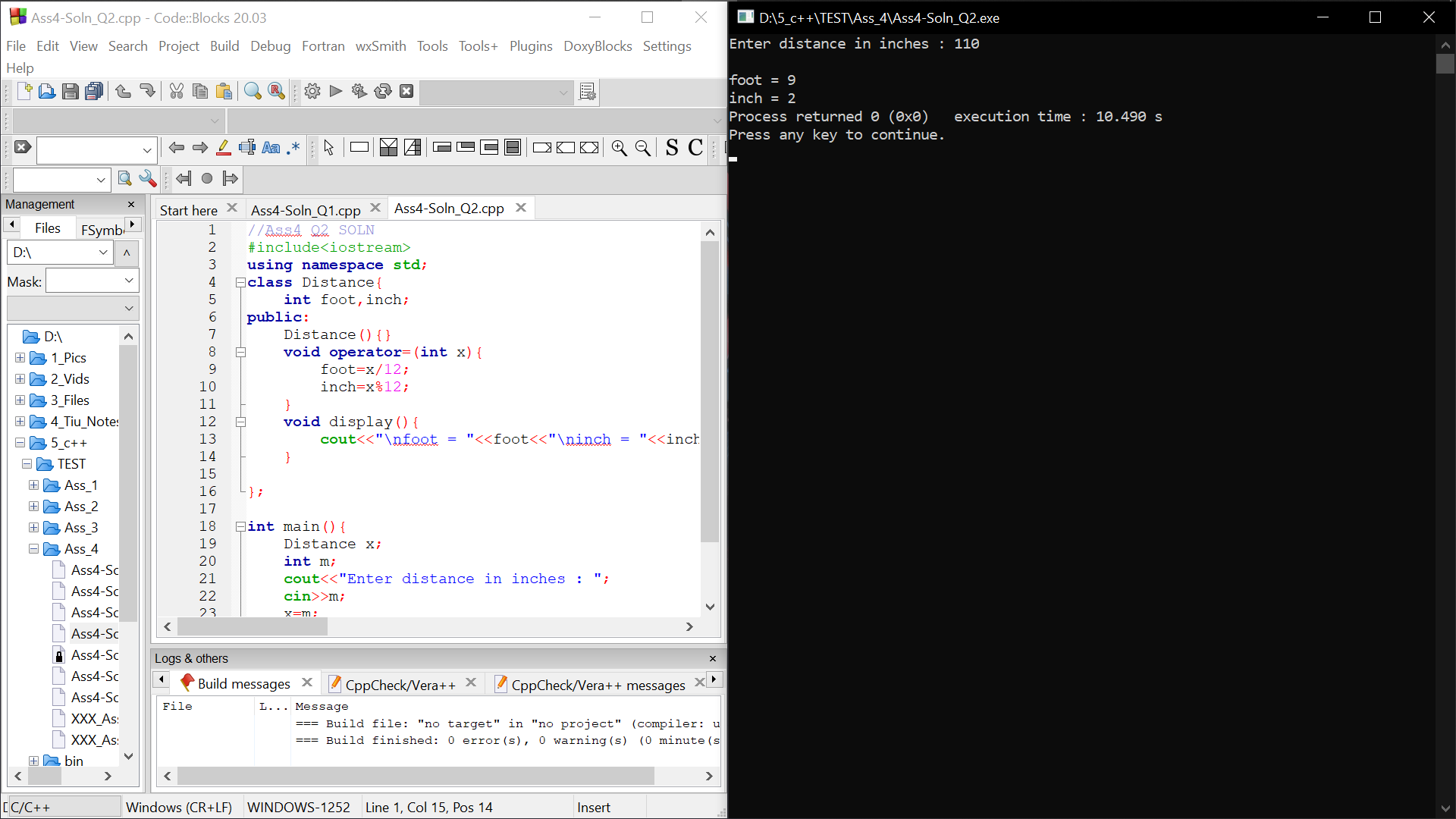
cin>>m;

x=m;

x.display();

}

**OUTPUT**



2. **Write a program to demonstrate the difference between**

***Assignment operator overloading Vs. Copy Constructor.***

**ANS: CODE**

#include<iostream>

using namespace std;

class Test{

int a,b;

public:

Test(int s,int t){

a=s;

b=t;

}

Test(Test &x){

a=x.a;

b=x.b;

}

void display(){

cout<<"a="<<a<<"\nb="<<b<<"\n";

}

void operator =(Test ob){

a=ob.a;

b=ob.b;

}

};

int main(){

Test ob(10,20);

Test ob1(1000,2000); //Creating Objects

cout<<"1st Object data\n";

ob.display();

cout<<"2nd Object data\n";

ob1.display();

cout<<"Using Copy Constructor...!!!\n";

Test ob2=ob; //Copy Constructor

ob2.display();

cout<<"Using Assignment Operator Overloading...!!!\n";

ob2=ob1; //Assignment Operator Overloading

ob2.display();

}

**OUTPUT**

