# EXPERIMENT NUMBER –Practical 4.1

STUDENT’S NAME – SHINDE SMITA SHAHAJI

STUDENT’S UID – 20BCS4643

CLASS AND GROUP –CSE-IOT(GROUP B)

SEMESTER – 2ND

# TOPIC OF EXPERIMENT –

# LEARN HOW TO USE CONSTRUCTOR AND DESTRUCTOR IN C++

# AIM OF THE EXPERIMENT –

# 

# WAP to find area of rectangle using constructor overloading. Also define destructor to delete the memory allocated to objects

FLOWCHART/ ALGORITHM-

Start

# Step 1→ Creating a header file for input output stream and define the context.

# Step 2 → Create a class named area rectangle and include the int variables- ln, br, area.

# 

# Step 3 → Create a class constructor having same name arearect without taking any arguments initialize the values of ln and br as 10, 10 and print the values of length and breadth.

# Step 4 → Create a class constructor having same name arearect taking arguments as int l, b and copy the values in ln and br and print the values of length and breadth.

# Step 5 → Create a member function calculate() for calculating the area of rectangle.

# Step 6→ Create a member function show () for printing the output of arae of rectangle.

# Step 7 → Declare a destructor ~araerect() for deallocating the memory allocated to the constructors.

# Step 8 → Declare the int main function. Create area rect class object r1 and another r2 object with 75, 40 as arguments passing in the area rect constructor

# Step 9 → Call calculate() member function for r1 and r2 objects.

# Step 10 → Call show() member function to print the output.

# Stop

# PROGRAM CODE-

# #include<iostream>

# using namespace std;

# class area

# {

# int a,l,b;

# public:

# area()

# {

# l=50;

# b=45;

# cout<<"Simple constructor called\n";

# cout<<"length="<<l<<"\nbreadth="<<b<<endl;

# }

# area(int x,int y)

# {

# l=x;

# b=y;

# }

# void calc();

# void print();

# ~area();

# };

# void area::calc()

# {

# a=l\*b;

# }

# void area::print()

# {

# cout<<"Area is : "<<a<<endl;

# }

# area::~area()

# {

# cout << "Object is being deleted" << endl;

# }

# int main()

# {

# int l,b;

# area a1;

# a1.calc();

# a1.print();

# cout<<"Enter length and breadth for parameterised constructor:\n";

# cin>>l>>b;

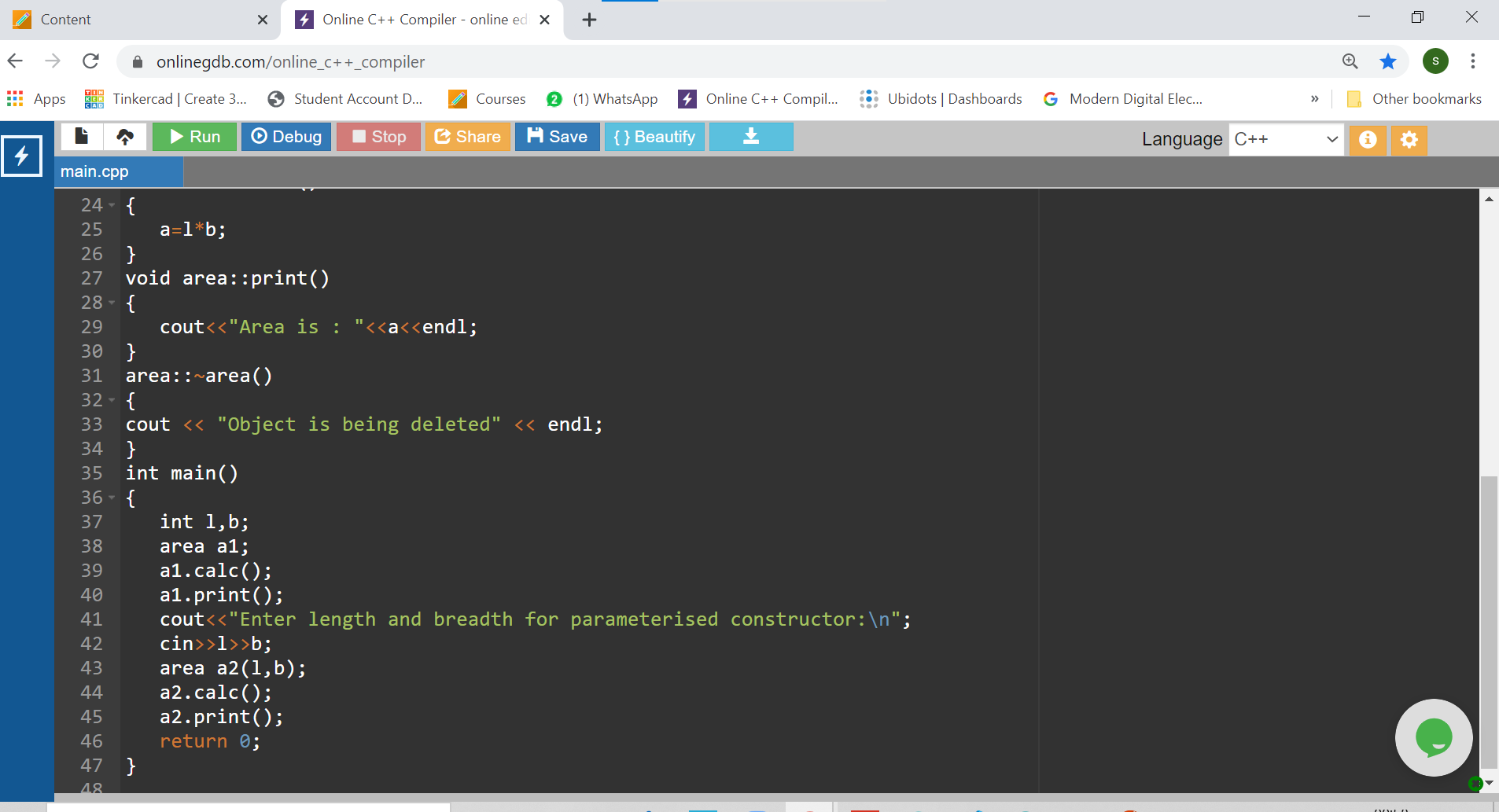
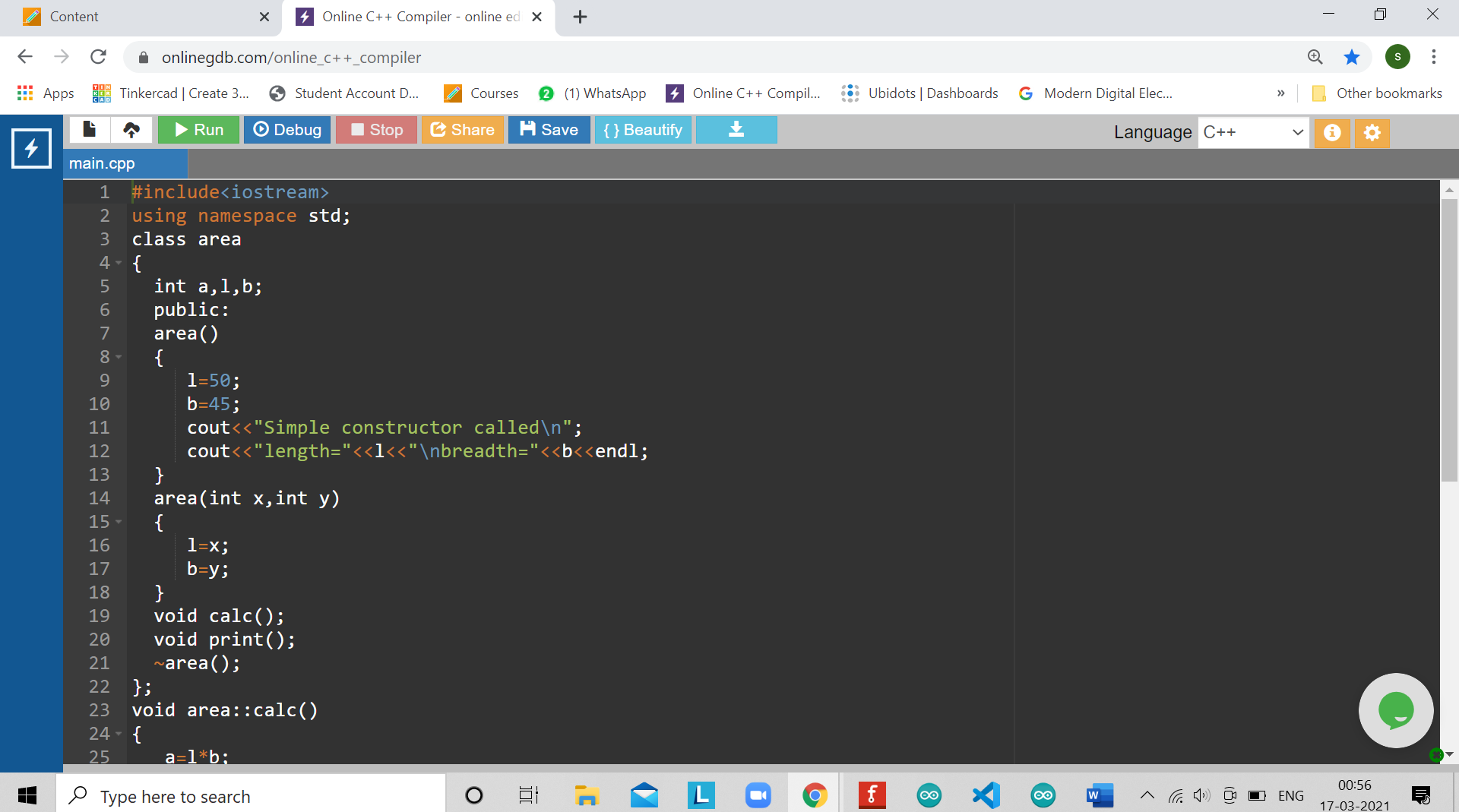
# area a2(l,b);

# a2.calc();

# a2.print();

# return 0;

# }



ERRORS ENCOUNTERED DURING PROGRAM’S EXECUTION

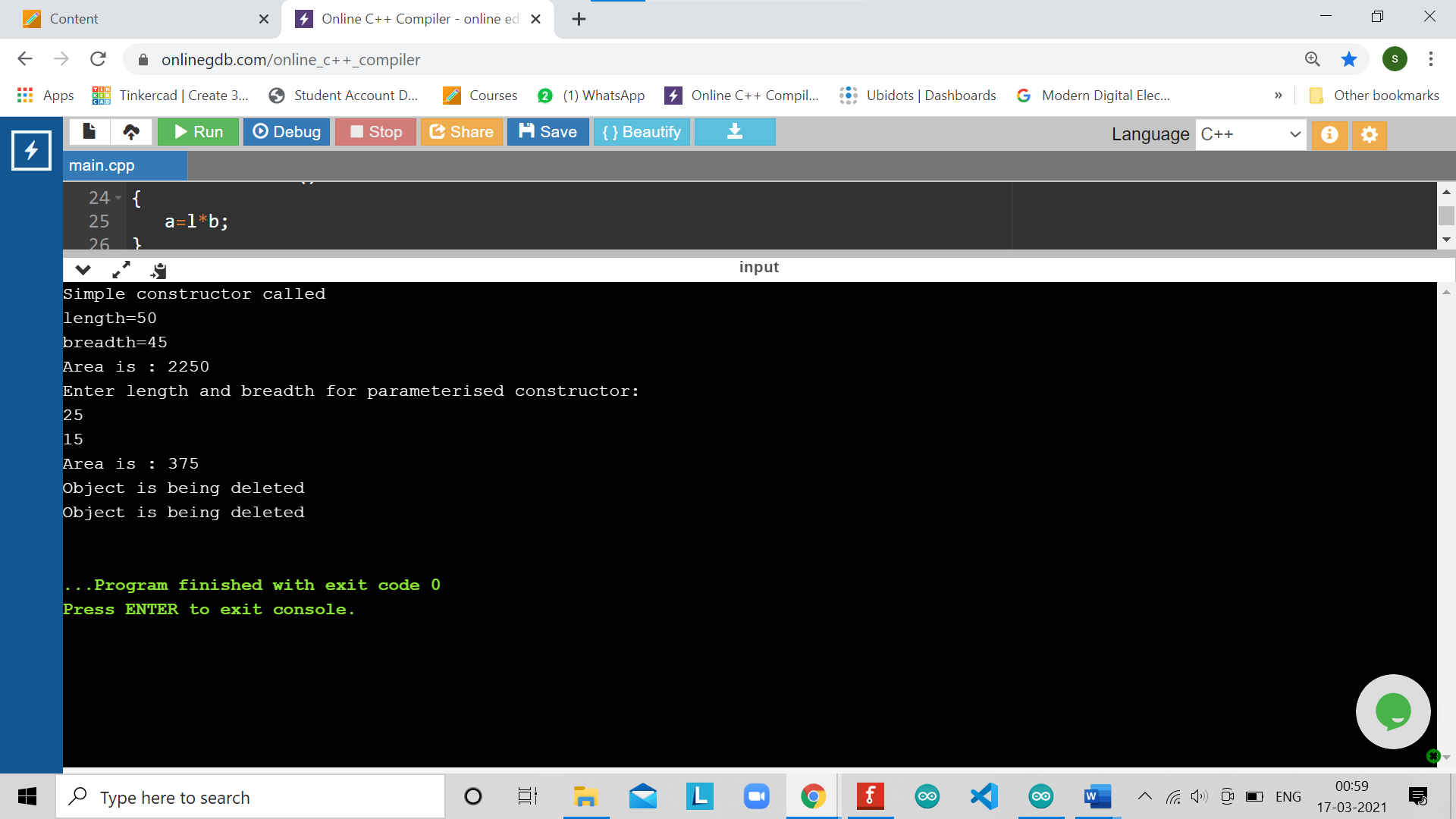
(Kindly jot down the compile time errors encountered)

No error Encountered

PROGRAMS’ EXPLANATION (in brief)

In this program we are finding the area of rectangle using default and parametrized constructor whose values of sides is entered by the user. Here the user first has to enter the dimensions of the rectangle and then is prompted with the result containing the area of the rectangle. Once the output is printed then the destructor is called. The destructor than clears the memory that was used by to program during runtime. Here we are using class constructor and destructor for calculating the area of rectangle and printing the output.

OUTPUT



# EXPERIMENT NUMBER –Practical EXPERIMENTAIM

STUDENT’S NAME – SHINDE SMITA SHAHAJI

STUDENT’S UID – 20BCS4643

CLASS AND GROUP –CSE-IOT(GROUP B)

SEMESTER – 2ND

# TOPIC OF EXPERIMENT –

# ****LEARN HOW TO USE CONSTRUCTOR AND DESTRUCTOR IN C++****

# AIM OF THE EXPERIMENT –

# WAP to create database of the following items: Name of the student (String), Roll number of the student (int), Height of the student (cm), Weight of the student (kg/gm).

# Create a Constructor to initialize values

# Create display () function to display the details

# Illustrate the use of copy constructor

# Also implement the concept of destructor.

FLOWCHART/ ALGORITHM-

Start

# Step 1→Creating a header file for input output stream and define the context.

# Step 2→ Create a class named student and include the int variables- uid, height, weight and also string variable name.

# Step 3→ Create a class constructor having same name student taking arguments of string n and int u, h, w, Copy the values of arguments in the declared variables.

# Step 4→ Create a copy constructor student and take argument as address of an object (&obj) and copy all the values inside the class variables.

# Step 5→Create a member function show () for printing the output of name, uid, height and weight of students.

# Step 6→ Declare a destructor ~student() for deallocating the memory allocated to the constructor.

# Step 7→ Declare the int main function. Create and initialize the student class objects s1 and s2.

# Step 8→ Call copy constructor with s1 and s2 as address of objects.

# Step 9→ Call show() member function to print the output.

# Stop.

PROGRAM CODE-

#include <iostream>

using namespace std;

class student

{

private:

string name;

int rollNo;

int height;

int weight;

public:

student(string n,int r,int h,int w)

{

name=n;

rollNo=r;

height=h;

weight=w;

}

//member function to print student's details

void display(void);

~student();

};

void student::display(void){

cout << "Student details:\n";

cout << "\nName:"<< name << ",\nRoll Number:" << rollNo << ",\nHeight:" << height << ",\nWeight:" << weight;

}

student :: ~student()

{

cout<<"\ndestructor is called\n";

}

int main()

{

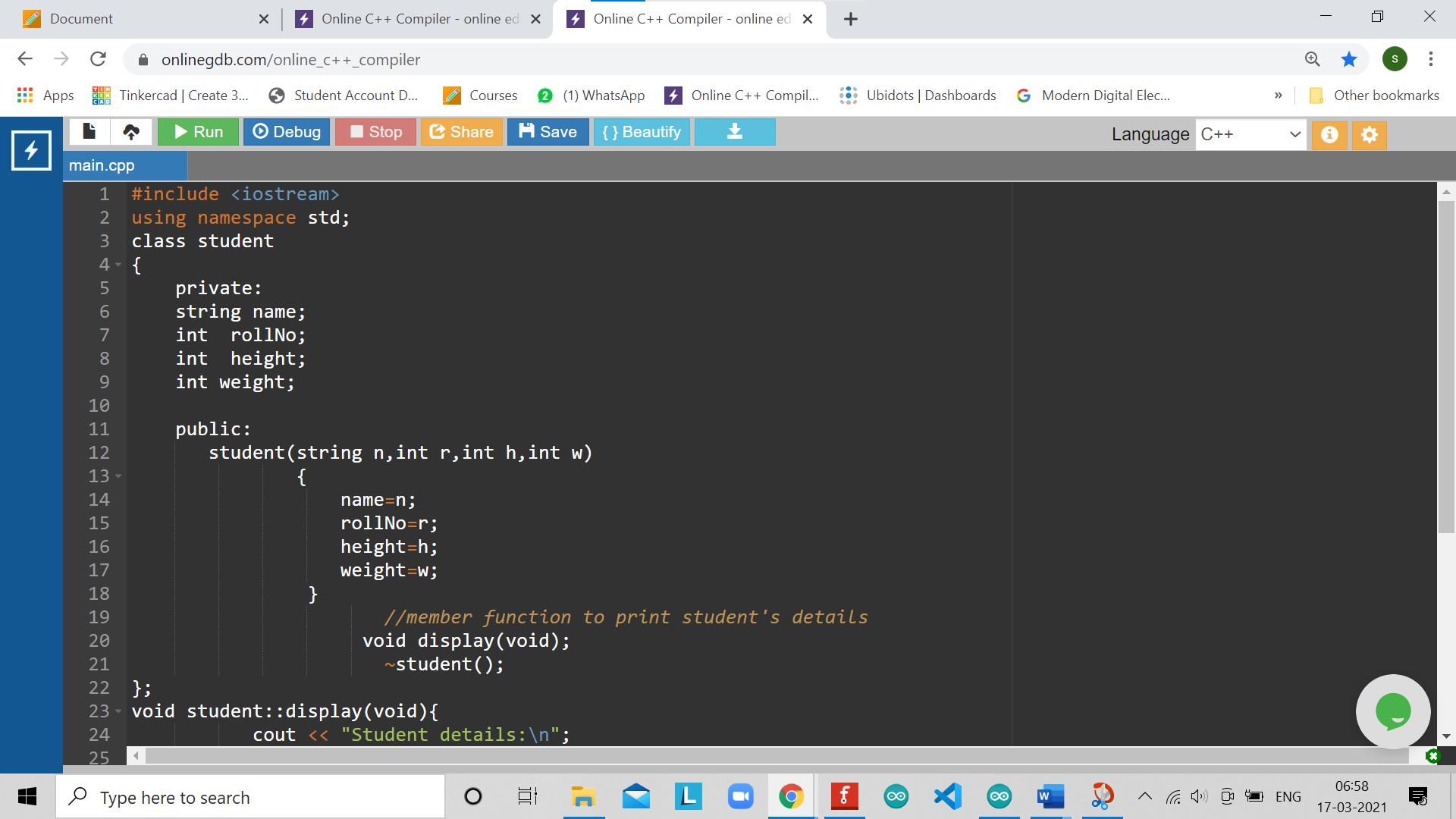
student std("Smita",4643,6,78); //object creation

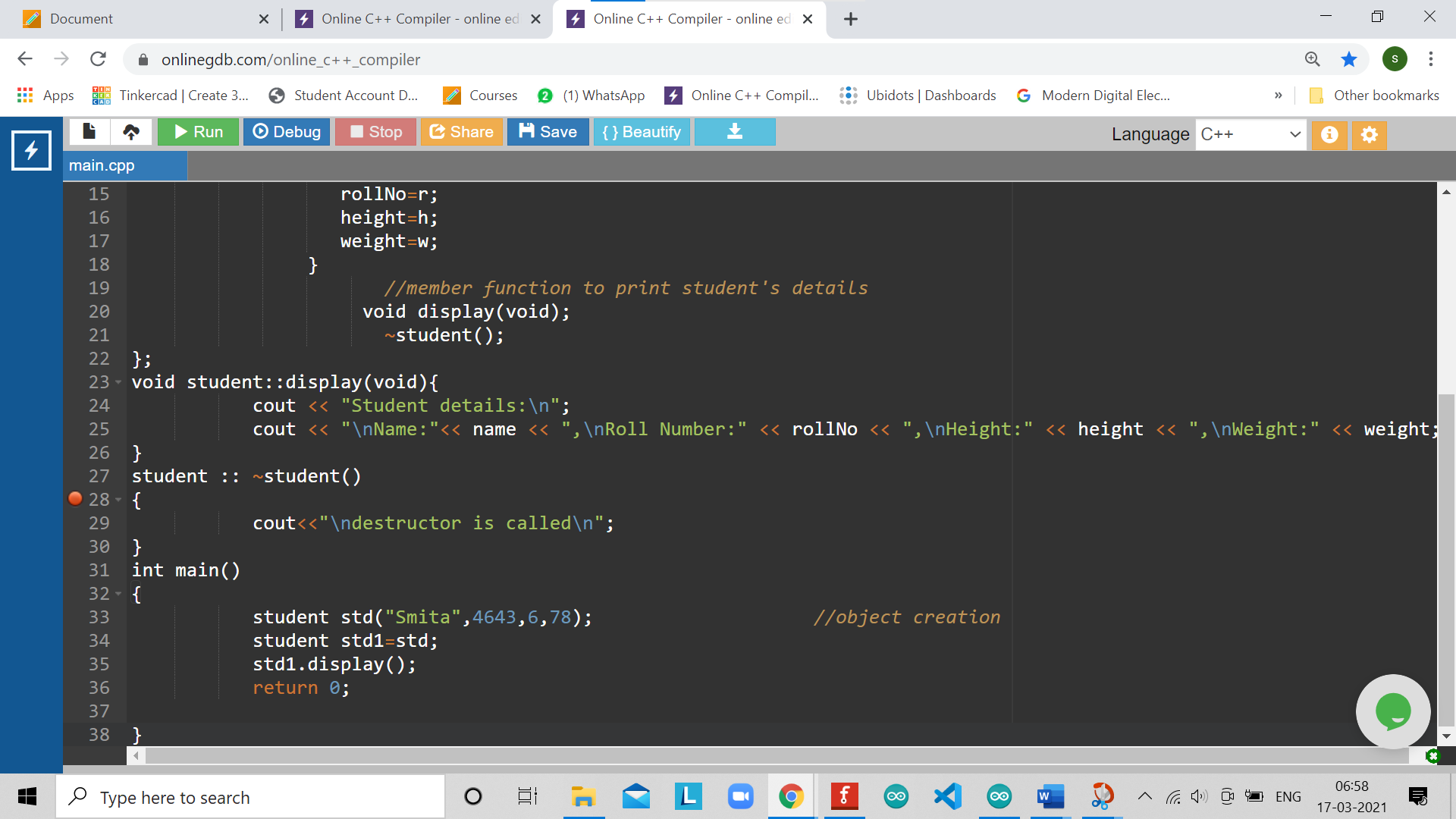
student std1=std;

std1.display();

return 0;

}





ERRORS ENCOUNTERED DURING PROGRAM’S EXECUTION

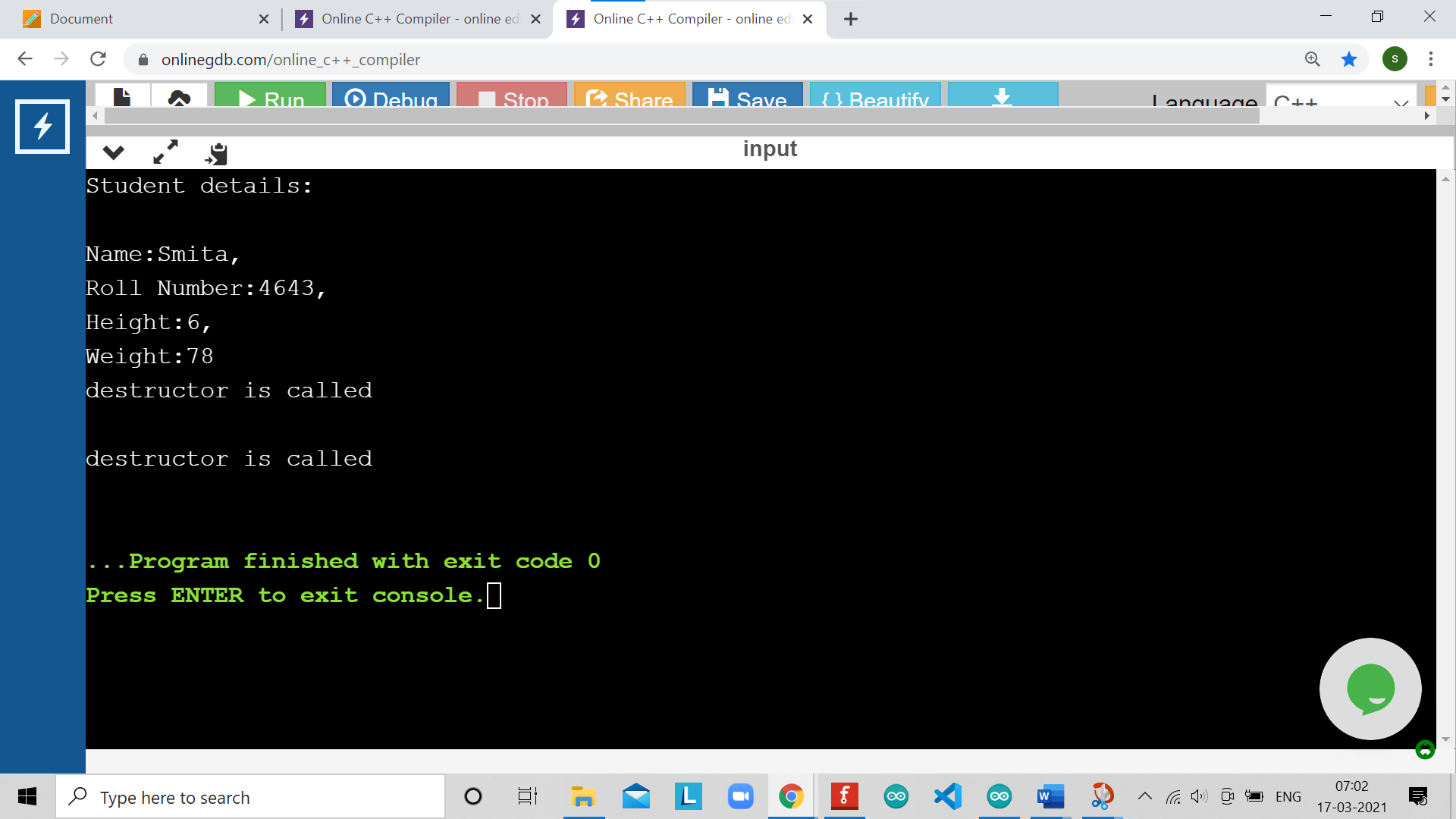
(Kindly jot down the compile time errors encountered)

No errors encountered

PROGRAMS’ EXPLANATION (in brief)

# In this program we are creating a database for name, roll number, height, weight of students and printing their values. Here we are using class constructor for taking input of variables, copy constructor and also destructor to delete the memory allocated to constructor.

OUTPUT



LEARNING OUTCOMES

|  |
| --- |
| * Identify situations where computational methods would be useful. |
| * Approach the programming tasks using techniques learnt and write pseudo-code. |
| * Choose the right data representation formats based on the requirements of the problem. |
| * Use the comparisons and limitations of the various programming constructs and choose the right one for the task. |

EVALUATION COLUMN (To be filled by concerned faculty only)

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Parameters** | **Maximum**  **Marks** | **Marks**  **Obtained** |
| 1. | Worksheet Completion including writing learning objective/ Outcome | 10 |  |
| 2. | Post Lab Quiz Result | 5 |  |
| 3. | Student engagement in Simulation/ Performance/ Pre Lab Questions | 5 |  |
| 4. | Total Marks | 20 |  |