

Basic Class Implementation

By

Jorge Jurado-Garcia

EE2510 Sec. 021, Spring 2021

Week 2 lab

Milwaukee School of Engineering

Submitted to:

Professor: Joshua D. Carl, Ph.D

EECS Department

Date Report Submitted: 03/18/21

Objective

The objective of this lab is to create two classes, class 1 that represents a rectangle and class 2 that represents a box. After creating a class, we then created a program that reads in an input file and uses the information in the text file to create objects. The objects should then print on console for the given results.

Description

We will have three input files named, input1, input2, and input3. This program will read the first line as the number rectangles to create line 2: number of boxes to create, and of the other lines finding the width, length, and height depending on if it has a B or R as the first character. The character “R” will find the variables to create a rectangle and “B” will find the variables to create a Box. The program will support any amount of number of boxes and rectangles for all real positive integers. The class names were given on a UML class diagram.

Conclusions

The lab was successful and was able to implement all the required operators detailed in LAB2_S21 pdf. What I learned during this lab is learning how to create a class declaration file. Which is a header file with the variables being used, functions prototype, and what information is public and private. From there I learned how to create a class definition .ccp file with all function’s information and “guts”. These two different files will work together to create an object when declared. The biggest problem I had with in this lab was not necessary creating the class but rather retrieving the data file and uses it for an specific object. I was able to solve this by learning what exactly the functions that fstream and iostream library do. And reading every element individual and each time determine what to do with the data. Later, when trying to finish this lab and different files with less code. I forgot that ifstream input(info,ios::in) creates an object input and reads what every file I tell it to inside the parathesis (ios::in). Therefore, I can create an object named input but read different files if I use a string variable named file and let the sure decide what file to open. Such user inserts “hello” my program will create a string named “hello.txt”. Overall, I learned a lot during a lot and was able to get it done in an appropriate time.

Main Code:

Main code:

```

/*
    main.cpp

    Created on: Mar 17, 2021
    Author: Jorge Jurado-Garcia

    Creating a program that reads an input file
    and uses the information in the text file to create objects
    should display the values stored in the data members,
    call member functions and display the results of the member functions

*/
#include <iostream>
#include <fstream>
#include <sstream>
#include <cstdlib> //exit function prototype

#include <string>
using namespace std;

#include "rectangle.h"
#include "MyBox.h"

int Request();
string convert(int number, string filename);

int main()
{
    //The program you write should be able to support any number of boxes and
    rectangles (from 0 to N), and
    //it should be able to process the file with the boxes and rectangles
    listed in any order.
    while(1){
        int request;
        cout << "Hello world!" << endl;

        //creating this string that can never be change
        string filename;
        filename = "input";

        //file name being returned
        string file;

        //the file the user wants to read is selected by this function
        request = Request();

        file = convert(request,filename);

        //ifstream constructor opens the file name
        //for this example we are opening a file named input and whatever integer
        precessed it
        ifstream input(file + ".txt",ios::in);
    }
}

```

```

    //reading information in from a file
    cout<<"Reading the file"<<endl;

    //data being collected type for characters
    string data;

    //open a file for inputs only using ios::in
    if(input.is_open() ){
        cout<<"your file input1 is open"<<endl;

        //data type being captured for line 1 and line 2
        int R_amount; //amount of rectangles needed
        int B_amount; //amount of boxes needed

        //these variables is how we will compare are objects made vs objects
        created
        int boxes = 0;
        int rect = 0;

        //reading line 1
        input>>R_amount;

        //reading line 2
        input>>B_amount;

        //notions for console
        cout<<"reading the file and determining the box and rectangle
        needed"<<endl;

        //creating the objects
        Rectangle empArr1[R_amount]; //dynamic memory
        //here we are creating an array of the amount of objects needed
        cout<<"creating "<<R_amount<<" rectangles"<<endl;

        MyBox empArr2[B_amount]; //dynamic memory
        cout<<"creating "<<B_amount<<" MyBoxes"<<endl;

        while( !input.eof() ){
            input>>data;
            cout<<"data in string: "<<data<<endl;
            //since the string always has the B or R in front we can use this
            if(data == "B"){
                //this where we can get are height,length,width for MyBox
                cout<<"This is Box"<<endl;

                //amount of integers in the line
                int num[3];

                for(int i=0; i<3; i++){ //read the values in the array

                    input>>num[i];
                }//end for

                //next we want to select the parts of the array and save it
                //as part length,width, and height for are object[s]
                empArr2[boxes].length = num[0];

```

```

empArr2[boxes].width = num[1];
empArr2[boxes].height = num[2];

cout<<"length: "<<empArr2[boxes].length<<endl;
cout<<"width: "<<empArr2[boxes].width<<endl;
cout<<"Height: "<<empArr2[boxes].height<<endl;

cout<< empArr2[boxes].getSurfaceArea()<<endl;
cout<<empArr2[boxes].getVolume()<<endl;

    if( boxes == B_amount){
        break;
    }//end if
    else{
        boxes = boxes+1;
    }//end else
} //end if
else if(data == "R"){
    //this is rectangle need length and width
    cout<<"This is a Rectangle"<<endl;
    int num[2];
    for(int i=0; i<2; i++){
        //reads the value as a number
        input>>num[i];
    }//end for

    //next we want to select the parts of the array and save it
    //as part length,width, and height
    empArr1[rect].length = num[0];
    empArr1[rect].width = num[1];
    cout<<"length: "<<empArr1[rect].length<<endl;
    cout<<"width: "<<empArr1[rect].width<<endl;
    cout<< empArr1[rect].getArea()<<endl;
    cout<< empArr1[rect].getPerimeter()<<endl;

    if( rect == R_amount){
        break;
    }//end if
    else{
        rect = rect+1;
    }//end else

} // end else if

} //end while

/*//just to verify that the object[s] where created
//we will do a for loop and send out the information again
for(int i = 0; i< B_amount; i++){
    cout<<"verifying the information printed below for each
Box"<<endl;

    cout<<"["<<empArr2[i].length<<","<<empArr2[i].width<<","<<empArr2[i].height<<
"]"<<endl;
        cout<<empArr2[i].getSurfaceArea()<<endl;
        cout<<empArr2[i].getVolume()<<endl;
    }//end for loop

```

```

        //problem some of the data gets into a different array because of
the rading functions takes
        //the last character in the file
        //will have to ask professor on how to fix this issue
        for(int i = 0; i< R_amount; i++){
            cout<<"verifying the information printed below for each
Rectangle"<<endl;

cout<<"["<<empArr1[i].length<<","<<empArr1[i].width<<"]"<<endl;
            cout<<empArr1[i].getArea()<<endl;
            cout<<empArr1[i].getPerimeter()<<endl;

            }//end for loop
        */
input.close();
    } //end if file is open
    else{
        //exit program if ifsream could not open
        cerr<<"file could not be opened"<<endl;
        exit(1);
    }
    cout<<"\n All Done,"<<endl;
}

}
int Request(){
    int value;
    cout<<"enter the file you want to read"<<endl
        <<"1-intput1.txt file"<<endl
        <<"2-input2.txt file"<<endl
        <<"3-input3.txt file"<<endl;
    cin>>value;
    return value;
}

string convert(int number,string filename){
    //this code is for converting any value into a string
    string file;
    stringstream ss;
    ss << number;
    string str = ss.str();
    file = filename+ str;
    cout<<file<<endl;
    return file;
}

```

Class Definitions

```

/*
 * RandB.cpp
 *
 * Created on: Mar 17, 2021
 * Author: Jorge Jurado-Garcia
 *
 * Defines the Rectangle and Box class
 *
 */

#include "MyBox.h"
#include "rectangle.h"

#include <iostream>
#include <string>
using namespace std;
//this holds all the functions guts form the two classes that we are using
Rectangle::Rectangle() {
    //called when object is being created
    cout<<"***In default constructor***"<<endl;
    length = 0;
    width = 0;
}

int Rectangle::getArea() {
    cout<<"your Area:"<<endl;
    return length*width;
}

int Rectangle::getPerimeter() {
    cout<<"your Perimeter:"<<endl;
    return (2*length)+(2*width);
}

MyBox:: MyBox() {
    //called when object is being created
    cout<<"***In default constructor***"<<endl;
    length = 0;
    width = 0;
    height =0;
}

int MyBox::getSurfaceArea() {
    cout<<"your SurfaceArea:"<<endl;
    return (2*length*width) + (2*length*height) + (2*height*width);
}

int MyBox::getVolume() {
    cout<<"your Volume:"<<endl;
    return (length*height*width);
}

```


Class Declarations

My Box

```
/*
 * MyBox.h
 *
 * Created on: Mar 17, 2021
 * Author: Jorge Jurado-Garcia
 *
 * Declares the Rectangle class
 */

#ifndef MYBOX_H_INCLUDED
#define MYBOX_H_INCLUDED

#include <string>
using namespace std;

class MyBox{
public:
//these are needed functions and variables given from UML box from
//kab2.pdf on canvas MSOE
    //variables
    int length;
    int width;
    int height;

    //functions
    MyBox();
    int getSurfaceArea();
    int getVolume();
};

#endif // MYBOX_H_INCLUDED
```

Rectangle:

```
/*
 * rectangle.h
 *
 * Created on: Mar 17, 2021
 * Author: Jorge Jurado-Garcia
 *
 * Declares the Rectangle class
 */

#ifndef RECTANGLE_H_INCLUDED
#define RECTANGLE_H_INCLUDED

#include <string>
using namespace std;

class Rectangle{
public:
//all of these functions and variables are being given from
//UML class diagram ~lab2.pdf on canvas MSOE
//this declares the variables being used
    int length;
    int width;

    //this declares the functions being used for this class
    Rectangle();
    int getArea();
    int getPerimeter();
};

#endif // RECTANGLE_H_INCLUDED
```


