EUROPEAN UNIVERSITY OF LEFKE

Faculty of Engineering

Department of Computer Engineering



COMP218

OBJECT-ORIENTED PROGRAMMING

**LAB WORK NO. 5**

Prepared by **David O. Ladipo** (174574)

Submitted to Dr. Ferhun Yorgancıoğlu

**Task-1:** Write a C++ program that implements the geometric shape “rectangle” as an object. For that, you should first specify all necessary properties of the object as data members, and then write all necessary member functions needed for the correct operation of the class. For each data member, you should provide a set and a get function. The class declaration should also include a constructor and a destructor. Remember that for a rectangle we can calculate the area, circumference, and diagonal length. All member functions should be defined within the class declaration.

#include <iostream>

#include <math.h>>

using namespace std;

class Rectangle {

private:

float length, breadth, area, circumference, diagonal;

inline float Area();

inline float Circumference();

inline float Diagonal();

public:

Rectangle();

Rectangle(float,float);

~Rectangle();

void setRectangle(float, float);

void setLength(float);

void setBreadth(float);

float getLength();

float getBreadth();

float getArea();

float getCircumference();

float getDiagonal();

};

//Constructors & Destructor

Rectangle::Rectangle()

{

setRectangle(length=0,breadth=0);

}

Rectangle::Rectangle(float l, float b)

{

setRectangle(l=0, b=0);

}

Rectangle::~Rectangle(){};

//Set functions

void Rectangle::setRectangle(float l, float b)

{

setLength(l);

setBreadth(b);

}

void Rectangle::setLength(float l)

{

length = l;

}

void Rectangle::setBreadth(float b)

{

breadth = b;

}

//Get functions

float Rectangle::getLength()

{

return length;

}

float Rectangle::getBreadth()

{

return breadth;

}

float Rectangle::getArea()

{

return Area();

}

float Rectangle::getCircumference()

{

return Circumference();

}

float Rectangle::getDiagonal()

{

return Diagonal();

}

//Data Member functions

inline float Rectangle::Area()

{

return area = length \* breadth;

}

inline float Rectangle::Circumference()

{

return circumference = 2\*(length+breadth);

}

inline float Rectangle::Diagonal()

{

return diagonal = sqrt((pow(length,2) + pow(breadth,2)));

}

void menu(){

cout<<"1: Set Rectangle"<<endl;

cout<<"2: Calculate Area"<<endl;

cout<<"3: Calculate Circumference"<<endl;

cout<<"4: Calculate Diagonal"<<endl;

cout<<"5: Exit Program"<<endl;

}

int main()

{

Rectangle r;

float l,b,ans;

int option;

menu();

while(1){

cout<<endl<<" [Choose any option from the MENU] "<<endl;

cin>>option;

switch(option){

case 1:

cout<<"Set a function"<<endl;

cout<<"Rectangle Length: ";

cin>>l;

cout<<"Rectangle Breadth: ";

cin>>b;

r.setLength(l);

r.setBreadth(b);

break;

case 2:

ans = r.getArea();

cout<<"Area of Rectangle is "<<ans<<" square."<<endl;

break;

case 3:

ans = r.getCircumference();

cout<<"Circumference of Rectangle is "<<ans<<" units."<<endl;

break;

case 4:

ans = r.getDiagonal();

cout<<"Diagonal of Rectangle is "<<ans<<" units."<<endl;

break;

case 5:

cout<<"Program Exited"<<endl;

return 0;

break;

default:

cout<<"Option is not available, Please select from the options above"<<endl;

break;

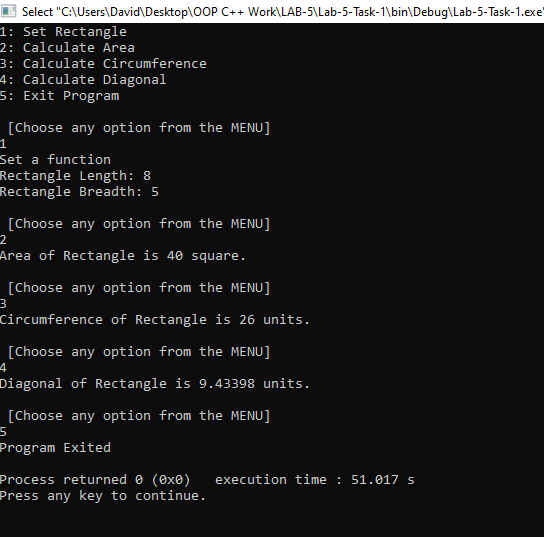
}

}

return 0;

}

**OUTPUT:**



**Task-2:** Consider the C++ program written in part-1 and rewrite it separating the *interface* from the *implementation*. Thus, the program should include a header file, an implementation file, and a driver program for testing.

**main.cpp**

#include <iostream>

#include "rectangle.h"

using namespace std;

void menu(){

cout<<"1: Set Rectangle"<<endl;

cout<<"2: Calculate Area"<<endl;

cout<<"3: Calculate Circumference"<<endl;

cout<<"4: Calculate Diagonal"<<endl;

cout<<"5: Exit Program"<<endl;

}

int main()

{

Rectangle r;

float l,b,ans;

int option;

menu();

while(1){

cout<<endl<<"\*\*\*\*\*\* |Choose any option from the MENU| \*\*\*\*\*\*"<<endl;

cin>>option;

switch(option){

case 1:

cout<<"Set a function"<<endl;

cout<<"Length: ";

cin>>l;

cout<<"Breadth: ";

cin>>b;

r.setLength(l);

r.setBreadth(b);

break;

case 2:

ans = r.getArea();

cout<<"Area of Rectangle is "<<ans<<" square."<<endl;

break;

case 3:

ans = r.getCircumference();

cout<<"Circumference of Rectangle is "<<ans<<" units."<<endl;

break;

case 4:

ans = r.getDiagonal();

cout<<"Diagonal of Rectangle is "<<ans<<" units."<<endl;

break;

case 5:

cout<<"Program Terminated"<<endl;

return 0;

break;

default:

cout<<"Option is not available, Please select from the options above"<<endl;

break;

}

}

return 0;

}

**rectangle.cpp**

#include <iostream>

#include <math.h>

#include "rectangle.h"

using namespace std;

Rectangle::Rectangle()

{

setRectangle(0,0);

}

Rectangle::Rectangle(float l, float b)

{

setRectangle(l, b);

}

Rectangle::~Rectangle(){};

//Set functions

void Rectangle::setRectangle(float l, float b)

{

setLength(l);

setBreadth(b);

}

void Rectangle::setLength(float l)

{

length = l;

}

void Rectangle::setBreadth(float b)

{

breadth = b;

}

//Get functions

float Rectangle::getLength()

{

return length;

}

float Rectangle::getBreadth()

{

return breadth;

}

float Rectangle::getArea()

{

return Area();

}

float Rectangle::getCircumference()

{

return Circumference();

}

float Rectangle::getDiagonal()

{

return Diagonal();

}

//Data Member functions

inline float Rectangle::Area()

{

return area = length \* breadth;

}

inline float Rectangle::Circumference()

{

return circumference = 2\*(length+breadth);

}

inline float Rectangle::Diagonal()

{

return diagonal = sqrt((pow(length,2) + pow(breadth,2)));

}

**rectangle.h**

#ifndef RECTANGLE\_H

#define RECTANGLE\_H

class Rectangle {

private:

float length, breadth, area, circumference, diagonal;

inline float Area();

inline float Circumference();

inline float Diagonal();

public:

Rectangle();

Rectangle(float,float);

~Rectangle();

void setRectangle(float, float);

void setLength(float);

void setBreadth(float);

float getLength();

float getBreadth();

float getArea();

float getCircumference();

float getDiagonal();

};

#endif // RECTANGLE\_H

