EUROPEAN UNIVERSITY OF LEFKE

Faculty of Engineering

# Department of Computer Engineering



# COMP218

OBJECT-ORIENTED PROGRAMMING

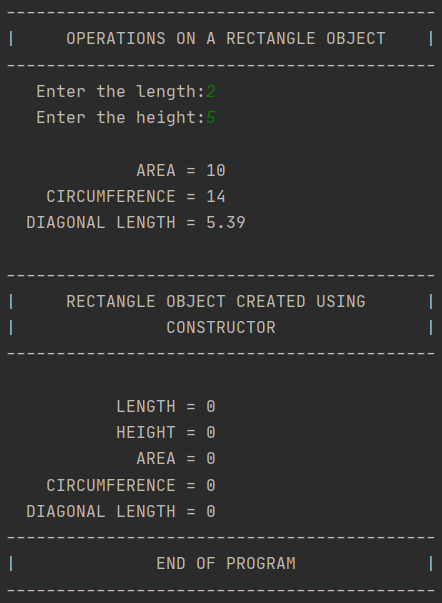
## Lab Work No. 6

Prepared by Seward Richard Mupereri (20140175)

Submitted to Dr. Ferhun Yorgancıoğlu

### Task (1)

#include <iostream>  
#include <iomanip>  
#include <cmath>  
  
using namespace std;  
  
class rectangle  
{  
public:  
 rectangle();  
 rectangle( int, int );  
 ~rectangle();  
 void setLength( int );  
 void setHeight( int );  
  
 int getLength();  
 int getHeight();  
 int getArea();  
 int getCircumference();  
 double getDLength();  
  
private:  
 int length, height;  
};  
  
rectangle::rectangle()  
{  
 setLength(0);  
 setHeight(0);  
}  
  
rectangle::rectangle( int l, int h )  
{  
 setLength( l );  
 setHeight( h );  
}  
  
rectangle::~rectangle(){};  
  
void rectangle::setLength( int l )  
{  
 length = l;  
}  
  
void rectangle::setHeight( int h )  
{  
 height = h;  
}  
  
int rectangle::getLength()  
{  
 return length;  
}  
  
int rectangle::getHeight()  
{  
 return height;  
}  
  
int rectangle::getArea()  
{  
 return ( getLength() \* getHeight() );  
}  
  
int rectangle::getCircumference()  
{  
 return ( ( getLength() + getHeight() ) \* 2 );  
}  
  
double rectangle::getDLength()  
{  
 return ( sqrt( pow( getLength(), 2 ) + pow( getHeight(), 2 ) ) );  
}  
  
int main()  
{  
 rectangle r;  
 int length, height;  
  
 cout << "-------------------------------------------" << endl;  
 cout << "| OPERATIONS ON A RECTANGLE OBJECT |" << endl;  
 cout << "-------------------------------------------" << endl;  
  
 cout << setw(20) << "Enter the length:";  
 cin >> length;  
 r.setLength( length );  
  
 cout << setw(20) << "Enter the height:";  
 cin >> height;  
 r.setHeight( height );  
  
 cout << endl;  
  
 cout << setw(20) << "AREA = " << r.getArea() << endl;  
 cout << setw(20) << "CIRCUMFERENCE = " << r.getCircumference() << endl;  
 cout << setw(20) << "DIAGONAL LENGTH = " << setprecision(3) << r.getDLength() << endl << endl;  
  
 cout << "-------------------------------------------" << endl;  
 cout << "| RECTANGLE OBJECT CREATED USING |" <<endl;  
 cout << "| CONSTRUCTOR |" << endl;  
 cout << "-------------------------------------------" << endl;  
  
  
 rectangle r1(0, 0);  
  
 cout << endl;  
  
 cout << setw(20) << "LENGTH = " << r1.getLength() << endl;  
 cout << setw(20) << "HEIGHT = " << r1.getHeight() << endl;  
 cout << setw(20) << "AREA = " << r1.getArea() << endl;  
 cout << setw(20) << "CIRCUMFERENCE = " << r1.getCircumference() << endl;  
 cout << setw(20) << "DIAGONAL LENGTH = " << setprecision(3) << r1.getDLength() << endl;  
  
 cout << "-------------------------------------------" << endl;  
 cout << "| END OF PROGRAM |" << endl;  
 cout << "-------------------------------------------" << endl;  
  
 return 0;  
}



### Task (2)

Header file – rectangle.h

#ifndef RECTANGLE\_H  
#define RECTANGLE\_H  
  
class rectangle  
{  
public:  
 rectangle();  
 rectangle( int, int );  
 ~rectangle();  
 void setLength( int );  
 void setHeight( int );  
  
 void getRectangle( rectangle );  
 int getLength();  
 int getHeight();  
 int getArea();  
 int getCircumference();  
 double getDLength();  
  
 void getX\_Base\_Cord ();  
  
private:  
 int length, height;  
};  
  
#endif //RECTANGLE

Implementation file – rectangle.cpp

#include <iostream>  
#include <iomanip>  
#include <cmath>  
#include "rectangle.h"  
  
using namespace std;  
  
rectangle::rectangle()  
{  
 setLength(0);  
 setHeight(0);  
}  
  
rectangle::rectangle( int l, int h )  
{  
 setLength( l );  
 setHeight( h );  
}  
  
rectangle::~rectangle(){};  
  
void rectangle::setLength( int l )  
{  
 length = l;  
}  
  
void rectangle::setHeight( int h )  
{  
 height = h;  
}  
  
void rectangle::getRectangle(rectangle r)  
{  
 int n = 25;  
 cout << setw(n) << "LENGTH = " << r.getLength() << endl;  
 cout << setw(n) << "HEIGHT = " << r.getHeight() << endl;  
 cout << setw(n) << "AREA = " << r.getArea() << endl;  
 cout << setw(n) << "CIRCUMFERENCE = " << r.getCircumference() << endl;  
 cout << setw(n) << "DIAGONAL LENGTH = " << setprecision(3) << r.getDLength() << endl;  
  
}  
  
int rectangle::getLength()  
{  
 return length;  
}  
  
int rectangle::getHeight()  
{  
 return height;  
}  
  
int rectangle::getArea()  
{  
 return ( getLength() \* getHeight() );  
}  
  
int rectangle::getCircumference()  
{  
 return ( ( getLength() + getHeight() ) \* 2 );  
}  
  
double rectangle::getDLength()  
{  
 return ( sqrt( pow( getLength(), 2 ) + pow( getHeight(), 2 ) ) );  
}  
  
void rectangle::getX\_Base\_Cord()  
{  
 cout << setw(17) << "x1:" << 0 << setw(8 ) << "y1:" << 0 << endl;  
 cout << setw(17) << "x2:" << length << setw(8 ) << "y2:" << 0 << endl;  
 cout << setw(17) << "x3:" << 0 << setw(8 ) << "y3:" << height << endl;  
 cout << setw(17) << "x4:" << length << setw(8 ) << "y4:" << height << endl;  
}

Driver program – main.cpp

#include <iostream>  
#include <iomanip>  
#include "rectangle.h"  
  
using namespace std;  
  
int main() {  
 rectangle r;  
 int length, height;  
  
 cout << "-------------------------------------------" << endl;  
 cout << "| OPERATIONS ON A RECTANGLE OBJECT |" << endl;  
 cout << "-------------------------------------------" << endl;  
  
 cout << setw(25) << "Enter the length:";  
 cin >> length;  
 r.setLength(length);  
  
 cout << setw(25) << "Enter the height:";  
 cin >> height;  
 r.setHeight(height);  
  
 cout << endl;  
  
 r.getRectangle( r );  
 cout << "-------------------------------------------" << endl;  
 cout << "| RECTANGLE OBJECT CREATED USING |" <<endl;  
 cout << "| CONSTRUCTOR |" << endl;  
 cout << "-------------------------------------------" << endl;  
  
 rectangle r1(3, 1);  
  
 cout << endl;  
  
 r1.getRectangle( r1 );  
  
 cout << "-------------------------------------------" << endl;  
 cout << "| OPTIONAL TASK |" << endl;  
 cout << "-------------------------------------------" << endl;  
  
  
 r1.getX\_Base\_Cord();  
 cout << endl;  
  
 cout << "-------------------------------------------" << endl;  
 cout << "| END OF PROGRAM |" << endl;  
 cout << "-------------------------------------------" << endl;  
  
 return 0;  
}

Output:

