

Department of Computing Science & Information Systems

CPSC 1181

Lab#5

June 12, 2019

Objectives:

- practice Inheritance and polymorphism
- learn abstract class
- create class hierarchy

Preparation:

Study class notes and example of sections Inheritance and polymorphism
Optional: Study chapter 9 of your text book

Due date:

Due Date: 11:00 PM on Monday October 14, 2019

Where to upload:

Zip your files into yourstudentID.zip where yourstudentID is your student number, and upload it to dropbox lab5 in D2L.

What do to:

Important Notes:

- You should use full power of Polymorphism and inheritance in this lab assignment. Your marker will mark you based on the quality of your design. A working program is a must, but quality of your code and using polymorphism is another essential part of this lab assignment.
- You should not use keyword instanceof in your code for this lab assignment.

This lab assignment consists of one abstract class(GeometricShape), four subclasses extending from abstract class (Rectangle, Square, Oval, and Circle), and class ShapeCollection with a main method.

A Rectangle is defined by a location (xTopLeft,yTopLeft) and dimension (w , h).

A Square is defined by a location xTopLeft,yTopLeft) and dimension (size).

An Oval is defined by its center (x,y) and the horizontal and vertical radius (a,b).

A Circle is defined by its center (x,y) and radius (r).

1. The abstract class GeometricShape is give as shown below:

```
abstract class GeometricShape{
    private int x;
    private int y;
    public GeometricShape(int x,int y){
        this.x = x;
```

```

        this.y = y;
    }
    public int getX(){
        return x;
    }
    public int getY(){
        return y;
    }
    abstract public double getArea();
    abstract public String toString();
}

```

The class contains two private instance fields (x,y) that either represents the top-left coordinates of the shape for Rectangle and Square classes, or center of the shapes for Oval and Circle classes.

GeometricShape contains a constructor, two accessors methods, and two abstract methods. The method area() returns the area of the shape, and toString() method returns the string representation of the object. (Refer to the sample output of the program for the required format).

2. class Rectangle is a subclass of GeometricShape, and class Square is a subclass of Rectangle class. Rectangle has a width and a height. Those values should be positive integers. Add appropriate accessors methods to the class Rectangle.

Square is a subclass of Rectangle and should only contain a new constructor and toString() methods as the Rectangle version of all other methods will be sufficient. Square and Rectangle should have an appropriate constructor for initializing instance data.

In Rectangle, be sure to implement the abstract methods of the parent class.

A rectangle and a square are defined by their upper left coordinates, width and height. A square is a rectangle with equal width and height.

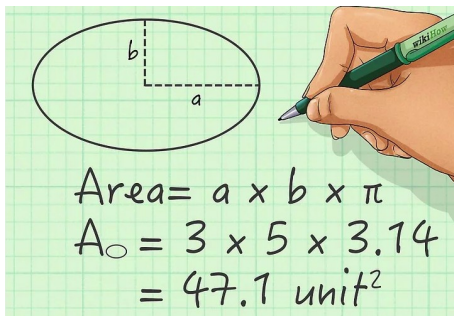
3. class Oval is a subclass of GeometricShape, and class Circle is a subclass of class Oval. Oval is defined by its center, a horizontal radius (a), and a vertical radius (b). Those values should be positive integers. Add appropriate accessors methods to the class Oval.

Circle is a subclass of Oval and it should only contain a new constructor and toString() method as the Oval version of other methods will be sufficient. Circle and Oval should have an appropriate constructor for initializing instance data.

In Oval, be sure to implement the abstract methods of the parent class.

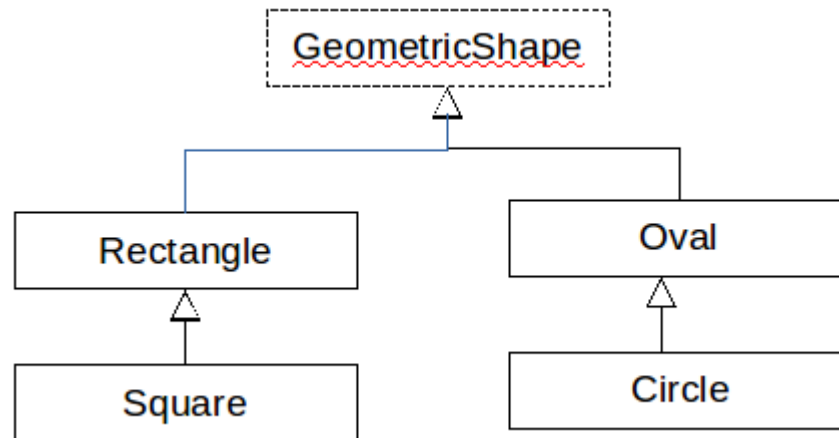
Note that a circle is an oval with equal horizontal and vertical radius (a=b).

The formula to find area of an oval is shown in the figure that follows:



source: <https://www.wikihow.com/Calculate-the-Area-of-an-Ellipse>

The UML diagram of the class hierarchy is shown below:



4. Download [ShapeCollection.java](#) first. The file contains all classes required for this lab assignment. Note that although all classes have package access level except class ShapeCollection which is public, all instance fields of the classes regardless of access level must be declared as private. The class contains a static main method to run the program, and a constructor, that creates an array of GeometricShape objects, and adds twenty random shapes with random parameters to the array.

5. Develop body of the four classes Rectangle, Square, Oval, and Circle.

6. Develop body of the following method of class ShapeCollections:

```
// calculates and returns the average of the area of the collection of the shapes of the array
public double findAverage( ) { }
```

```
// Returns reference of the shape with maximum area.
```

```
public GeometricShape getMax() { }
```

```
// Sort the shapes in the array in ascending order based on their area value
```

```
public void sort() { }
```

7. Uncomment the codes in the body of the main method to run the test cases. The format of the output of the test cases should be the same as shown in sample run of the program. The sample output of the test cases is shown below. Be sure that the program you are developing produces similar output. Study the format of the output and design your code based on the sample provided.

Note: You can use Math.round(..) method to round the values to the nearest integer for display purpose.

Test case 1:

average: 2223

Test case 2:

Square: [x:49, y:32, length:91, area:8281.0]

Test case 3:

list of the shapes:

Circle: [x:42, y:30, radius:9, area:254.0]
Square: [x:21, y:40, length:32, area:1024.0]
Square: [x:49, y:32, length:91, area:8281.0]
Square: [x:33, y:3, length:46, area:2116.0]
Square: [x:26, y:39, length:36, area:1296.0]
Circle: [x:8, y:19, radius:19, area:1134.0]
Rectangle [x:48, y:17, width:73, height:26, area:1898.0]
Oval: [x:14, y:18, h_radius:27, v_radius:56, area:4750.0]
Square: [x:22, y:31, length:57, area:3249.0]
Rectangle [x:24, y:38, width:11, height:10, area:110.0]
Oval: [x:25, y:47, h_radius:10, v_radius:43, area:1351.0]
Rectangle [x:27, y:1, width:56, height:40, area:2240.0]
Square: [x:29, y:48, length:25, area:625.0]
Circle: [x:14, y:37, radius:26, area:2124.0]
Rectangle [x:48, y:15, width:68, height:1, area:68.0]
Oval: [x:48, y:45, h_radius:1, v_radius:49, area:154.0]
Circle: [x:28, y:30, radius:31, area:3019.0]
Rectangle [x:22, y:14, width:99, height:80, area:7920.0]
Circle: [x:4, y:45, radius:30, area:2827.0]
Square: [x:39, y:47, length:4, area:16.0]

Test case 4:

shapes sorted in ascending order based on their area:

Square: [x:39, y:47, length:4, area:16.0]
Rectangle [x:48, y:15, width:68, height:1, area:68.0]
Rectangle [x:24, y:38, width:11, height:10, area:110.0]
Oval: [x:48, y:45, h_radius:1, v_radius:49, area:154.0]
Circle: [x:42, y:30, radius:9, area:254.0]
Square: [x:29, y:48, length:25, area:625.0]
Square: [x:21, y:40, length:32, area:1024.0]
Circle: [x:8, y:19, radius:19, area:1134.0]
Square: [x:26, y:39, length:36, area:1296.0]
Oval: [x:25, y:47, h_radius:10, v_radius:43, area:1351.0]
Rectangle [x:48, y:17, width:73, height:26, area:1898.0]
Square: [x:33, y:3, length:46, area:2116.0]
Circle: [x:14, y:37, radius:26, area:2124.0]
Rectangle [x:27, y:1, width:56, height:40, area:2240.0]
Circle: [x:4, y:45, radius:30, area:2827.0]
Circle: [x:28, y:30, radius:31, area:3019.0]
Square: [x:22, y:31, length:57, area:3249.0]
Oval: [x:14, y:18, h_radius:27, v_radius:56, area:4750.0]
Rectangle [x:22, y:14, width:99, height:80, area:7920.0]
Square: [x:49, y:32, length:91, area:8281.0]

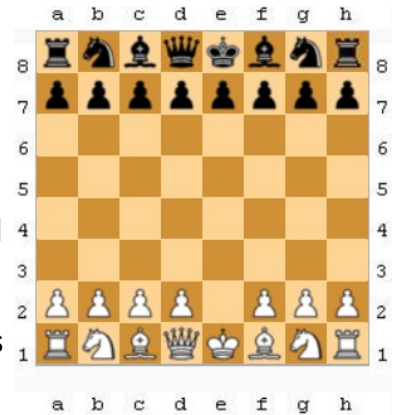
Bonus Part: [10 marks]

If you are interested in challenge, then this part is for your. You will have up to 10 bonus marks and you will learn lots from this part. I encourage all to try this part.

In this part you are going to create a chess board with chess pieces in their initial positions. If you develop this part, then you have a chance to gradually develop a fully chess program in the assignment that comes.

Chess is a recreational and competitive game for two players. Sometimes called Western Chess or International Chess to distinguish it from its predecessors and other chess variants, the current form of the game emerged in Southern Europe in the second half of the 15th century after evolving from similar, much older games of Indian origin.

Today, chess is one of the world's most popular games, played by millions of people worldwide in clubs, on line, by correspondence, in tournaments and informally. Aspects of art and science are found in chess composition and theory. Chess is also advocated as a way of enhancing mental powers.



What to do:

Download [chess_ver1.zip](#) file, and study it. The file contains six classes:

1. class Chess

Chess class contains the main method. Creates an instance of class GameController and starts the game, and passes control to an instance of GameController.

2. class GameController

GameController class contains a instance field of ChessBoard. Its job is to start the game, move the pieces, validate the move, and then use ChessBoard to move the piece. It also provides message if the move is not valid.

(We are not going to create a full chess program in the first version, so you do not need to modify it in this assignment. In fact the class GameController is developed for you and use it as is for this bonus part).

3. class ChessBoard

ChessBoard class is the the board of the game and has an $n \times n$ array each holds a unique piece in the game board position.

ChessBoard provides the following public methods :

```
// place each piece in the initial starting positions
public void populate();
```

```
//draw the board and the pieces
public void draw(Graphics2D g2)
```

Some of these methods are either fully or partially developed. Note that you may need other public or private methods to complete this class.

4. class Def

Def class contains all static fields. Its purpose is to define final static fields that can be used by all classes. Instead of defining final static fields in each classes, you can put them all in a single class and share among them. This is how we should manage constants in a real project. This class is partially developed.

5. class abstract Piece :

abstract class Piece is the base class for chess pieces such as Pawns, Kings, Rook, Bishop, Queen, and Knight. When creating an abstract class, you should always ask yourself following questions:

- What are the properties of individual pieces?
- Which methods should be declared as abstract?
- What is common to all game pieces?

Each piece should print itself.

Each piece has a color.

Each piece should report its type and color.

Some of them should be declared as abstract, and others as a regular method. This class is fully developed. You do not need to add code to it, unless you come with a better design.

6. class Pawn :

Pawn class is extended from abstract class Piece. It stands for Pawn on the chess board.

Classes you should develop

Develop other pieces of chess game, King, Queen, Rook, Bishop, and Knight.

7. class Dummy :

Develop class Dummy by extending class Piece. Dummies are just used as a place holder, they can reduce complexity of your program. How?

Why should I create Dummy object that does nothing?

What is the advantage of having dummies in a game board?

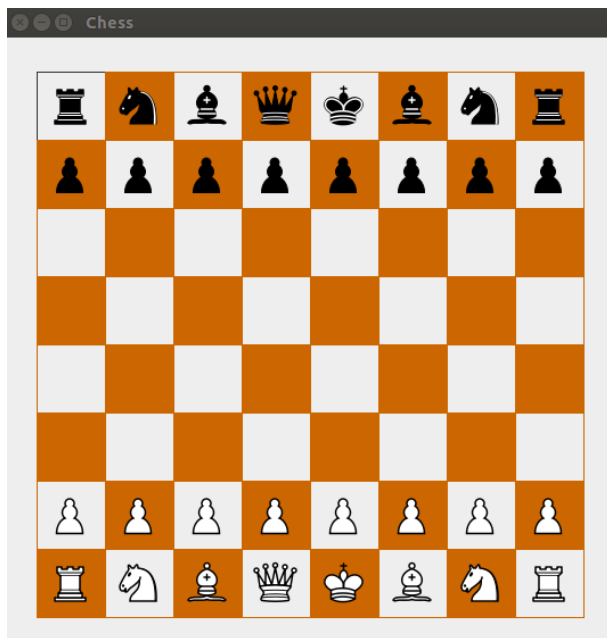
Discuss these questions with your classmates. Is there a better alternative for this?

You do not need to provide answer to the questions asked in the preceding paragraph. Just discuss and think about them.

8. Complete the program, place each chess piece in its initial position, and fill the empty squares with dummy objects.

Note that you should use full power of Polymorphism and inheritance in this lab assignment. Your marker will mark you based on the quality of your design. A working program is a must, but quality of your code and using polymorphism is another essential part of this lab assignment.

The initial state of the board is shown below:



What to submit

1. Comment all your classes and methods using the javadoc notation.
2. Capture and include the sample output of your program.
3. If you are doing the bonus part, then you need to create two folders, one for your lab assignment and one for the bonus part.
4. If you are using IDEs like Eclipse, then be sure that your program is compiled and run in command window, otherwise your lab assignment may not be marked.
5. Comments about your assignment if needed these comments are not the comments documenting your code but rather something you need to convey to us about your assignment
6. Zip your assignment into yourStudentId.zip file and submit it to D2l Dropbox: lab5

TOTAL MARK: 50 +10 Bonus