LAB 5

Exercise 1: Create a Class & Methods

- Create a circle class that
  - contains attributes radius, and a constant pi (3.14159)
  - has the following method
    - setRadius(double r)
    - double getRadius()
    - double getArea()
    - double getPerimeter()

Exercise 1

- Create a class header

```
public class Circle{



}
```

- Declare and create variable and constant of the class (radius and PI)

```
public class Circle{
        double radius;
        final double PI = 3.14159;


}
```

Exercise 1

- Create methods

```
public class Circle{
        double radius;
        final double PI = 3.14159;
        public void setRadius(double r){
                radius = r;
        }
        public double getRadius(){
                return radius;
        }
        public double getArea(){
                double area = PI*radius*radius;
                return area;
        }
        public double getPerimeter(){
                return 2*PI*radius;
        }
}
```

Exercise 2: Create a Driver Class

- Create a driver class named “CircleCaller”

```
public class CircleCaller{

        public static void main(String args){

        }

}
```

Exercise 3: Create a Circle Object

- Create a circle object with variable name c1

```
public class CircleCaller{
        public static void main(String args){
                Circle c1 = new Circle();
        }

}
```

Exercise 4: Use methods

- Create a circle object with variable name c1

```
public class CircleCaller{
        public static void main(String[] args){
                Circle c1 = new Circle();
                c1.setRadius(2.5);
                c1.getRadius();
                c1.getArea();
                c1.getPerimeter();
        }

}
```

Exercise 4

```
public class CircleCaller{
        public static void main(String[] args){
                Circle c1 = new Circle();
                c1.setRadius(2.5);
                System.out.println(c1.getRadius());
                System.out.println(c1.getArea());
                System.out.println(c1.getPerimeter());
        }

}
```

Exercise 4

- What about trying to print the method which has no return value?

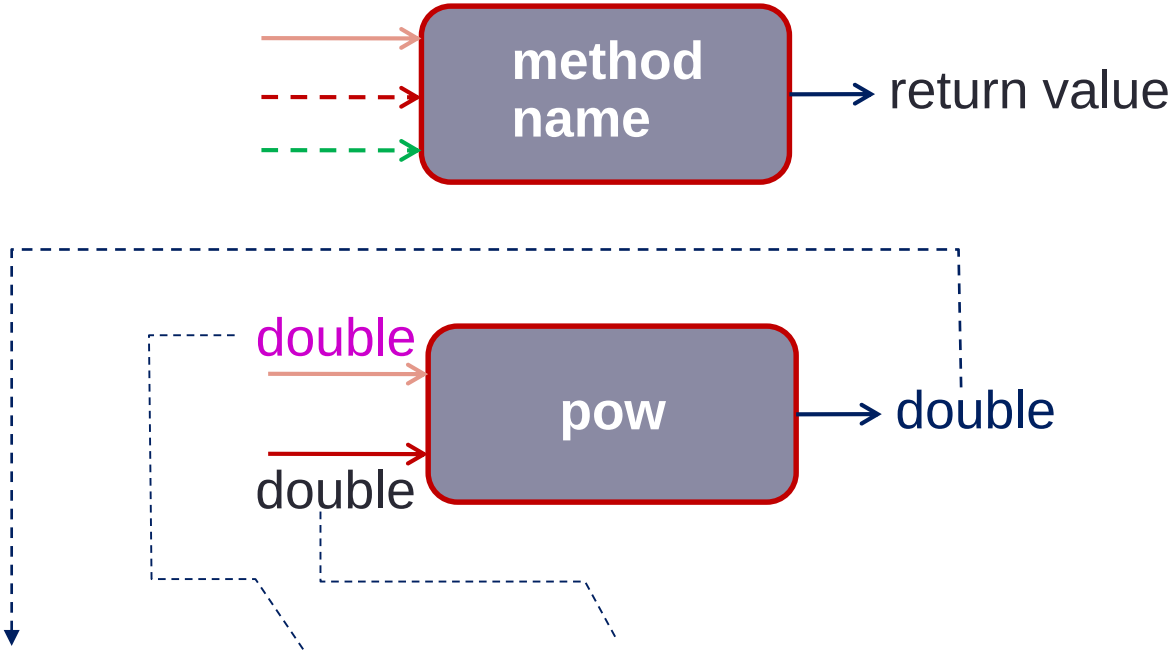
```
public class CircleCaller{
        public static void main(String[] args){
                Circle c1 = new Circle();
                System.out.println(c1.setRadius(2.5));
                System.out.println(c1.getRadius());
                System.out.println(c1.getArea());
                System.out.println(c1.getPerimeter());
        }

}
```

- Error

Exercise 5: Method Pattern

- Pattern 1



static double pow (double a, double b)

Returns the value of the first argument raised to the second argument.

Exercise 5

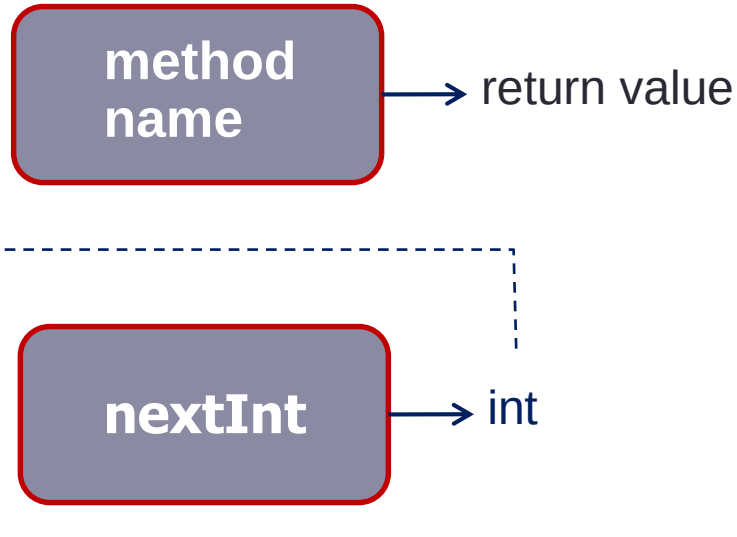
- Write a method to calculate area of *n* circles, where *n* is a number of circles as the input of the method

```
public class Circle{
        double radius;
        final double PI = 3.14159;
        public double getAreaOfCircles(int n){
                return n*PI*radius*radius;
        }
}
```

```
public class Circle{
        double radius;
        final double PI = 3.14159;
        public double getAreaOfCircles(int n){
                return n*getArea();
        }
}
```

You can user any method within the class without creating an object.

Exercise 5



public int nextInt()

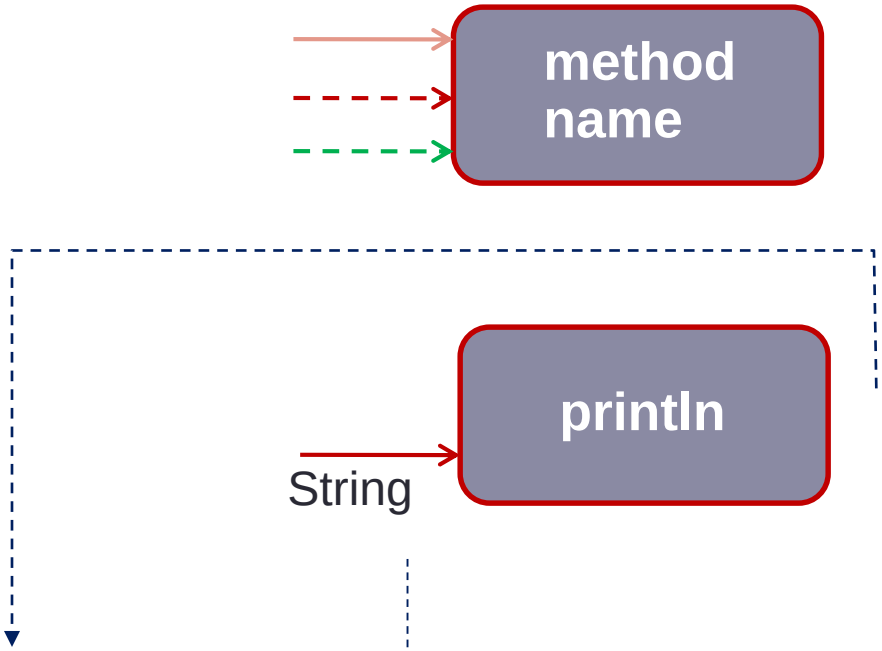
Scans the next token of the input as an int.

Exercise 5

- Write a method to calculate a diameter of the circle

```
public class Circle{
        double radius;
        final double PI = 3.14159;
        public double getDiameter(){
                return 2*radius;
        }
}
```

Exercise 5



public void println (String x)

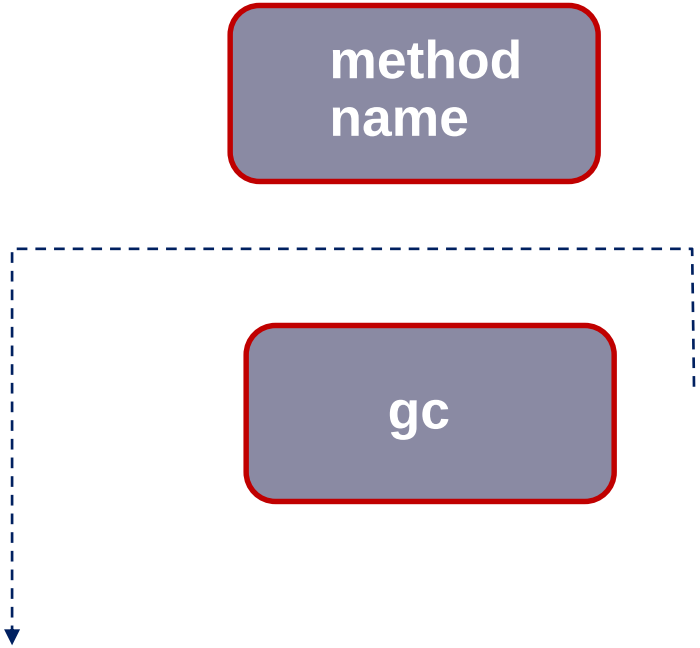
Prints a String and then terminate the line. This method behaves as though it invokes print(String) and then println().

Exercise 5

- Write a method to set a center of the circle by taking the input of coordination x and y.

```
public class Circle{
        double radius;
        final double PI = 3.14159;
        double xcoor;
        double ycoor;
        public void setCenter(int x, int y){
                xcoor = x;
                ycoor = y;
        }
}
```

Exercise 5



public static void gc ()

Runs the garbage collector.

Exercise 5

- Write a method to print the information of the circle (e.g., radius, area, coordinate)

```
public class Circle{
        double radius;
        final double PI = 3.14159;
        double xcoor;
        double ycoor;
        public void printInfo(){
                System.out.println("The circle have:");
                System.out.println("radius = "+radius);
                System.out.println("area = "+getArea());
                System.out.println("diameter = "+getDiameter());
                System.out.println("coordinate  = ("+xcoor+","+ycoor+")");
        }
}
```

- The Circle class (completed)

```
public class Circle{
            double radius;
            final double PI = 3.14159;
            double xcoor;
            double ycoor;
            public void setRadius(double r){
                        radius = r;
            }
            public double getRadius(){
                        return radius;
            }
            public double getArea(){
                        double area = PI*radius*radius;
                        return area;
            }
            public double getPerimeter(){
                        return 2*PI*radius;
            }

            public double getAreaOfCircles(int n){
                        return n*getArea();
            }
            public double getDiameter(){
                        return 2*radius;
            }
            public void setCenter(int x, int y){
                        xcoor = x;
                        ycoor = y;
            }
            public void printInfo(){
                        System.out.println("The circle have:");
                        System.out.println("radius = "+radius);
                        System.out.println("area = "+getArea());
                        System.out.println("diameter = "+getDiameter());
                        System.out.println("coordinate  = ("+xcoor+","+ycoor+")");
            }
}
```

Exercise 6

- Let’s add a static variable and method

```
public class Circle{
        double radius;
        final double PI = 3.14159;
        double xcoor;
        double ycoor;
        public static int countCircleObject;
        public void setRadius(double r){
                radius = r;
                countCircleObject++;
        }
        public static int getNumberOfCircleObjects(){
                return countCircleObject;
        }

        …
        …
        …
}
```

- Let’s modify your CircleCaller class as follows:

```
public class CircleCaller{
        public static void main(String[] args){
                Circle c1 = new Circle();
                c1.setRadius(2.5);
                System.out.println("---After the first Circle created---");
                System.out.println(c1.getNumberOfCircleObjects());
                System.out.println(Circle.getNumberOfCircleObjects());
                System.out.println(c1.countCircleObject);
                System.out.println(Circle.countCircleObject);

                Circle c2 = new Circle();
                c2.setRadius(4);
                System.out.println("---After the second Circle created---");
                System.out.println(c2.getNumberOfCircleObjects());
                System.out.println(Circle.getNumberOfCircleObjects());
                System.out.println(c2.countCircleObject);
                System.out.println(Circle.countCircleObject);

                Circle c3 = new Circle();
                System.out.println("---After the third Circle created---");
                System.out.println(c3.getNumberOfCircleObjects());
                System.out.println(Circle.getNumberOfCircleObjects());
                System.out.println(c3.countCircleObject);
                System.out.println(Circle.countCircleObject);
        }

}
```

Exercise 6

- Observe the result of the driver class – CircleCaller

Problems @ Javadoc Declaration Console

```
<terminated> CircleCaller [Java Application] C:\Program Files\Java\jre1.8.0_91\bin\javaw.exe (Oct 5, 2016, 1:22:16 PM)
---After the first Circle created---
1
1
1
1
---After the second Circle created---
2
2
2
2
---After the third Circle created---
2
2
2
2
```

- What do you learn?

Exercise 7

- Create a new class called CircleService to provide services for calculating area, diameter, and perimeter of any giving inputs of radius
- In this case, we will make use of a static modifier

Exercise 7

- Create a CircleService class with static methods

```
public class CircleService{
        private final double PI = 3.14159;
        public static void getArea(double radius){
                //put your code here
        }
        public static void getDiameter(double radius){
                //put your code here
        }
        public static void getPerimeter(double radius){
                //put your code here
        }
}
```

Exercise 7

- Create a CircleService class with static methods

```
public class CircleService{
        private final static double PI = 3.14159;
        public static double getArea(double radius){
                return PI*radius*radius;
        }
        public static double getDiameter(double radius){
                return 2*radius;
        }
        public static double getPerimeter(double radius){
                return 2*PI*radius;
        }
}
```

- Try to remover static modifier from PI variable, is there any errors?

Exercise 7

- Create a new CircleServiceCaller class

```
public class CircleServiceCaller{
        public static void main(String[] args){
                double r = 2.0;
                double area = CircleService.getArea(r);
                double diameter = CircleService.getDiameter(r);
                double perimeter = CircleService.getPerimeter(r);
                System.out.println("Area: "+area);
                System.out.println("Diameter: "+diameter);
                System.out.println("Perimeter: "+perimeter);
        }
}
```

- Run and observe result

DIY

- Create a Rectangle class that contains the following methods:
  - setWidthHeight(double w, double h) // set width and height of a rectangle
  - getPerimeter() // return the perimeter of a rectangle
  - getArea() // return the area of a rectangle
  - printInfo() //print width and height of the rectangle, print the perimeter and also its area

- Create a RectangleCaller class and in the class provide a main method that
  - Creates 3 rectangle objects
  - Sets the width and height of each object respectively: (5, 10), (2.5, 1.5), (25, 5)
  - Calls the other remaining methods provided in the Rectangle class

DIY

- Create a RectangleService class with the following static methods:
  - getPerimeter(double w, double h) // return the perimeter of a rectangle with given width (w) and height (h)
  - getArea(double w, double h) // return the area of a rectangle with given width (w) and height (h)

- Create a RectangleServiceCaller class and in the class provide a main method that
  - Prints a perimeter of a rectangle with width = 15, and height = 10
  - Prints an area of a rectangle with width = 8, and height = 7
  - Note that it must use the method in the RectangleService without creating any object !