LAB 8 Part I ITERATIONS

Objective

Computer scientists have described two general mechanisms for repeating steps: ‘recursion’ and ‘iteration’. Recursion works by having a method call itself again and again until the process is computed. Iteration provides a more explicit mechanism for repetition, although one that is slightly harder to analyze for correctness.

Most imperative languages provide three kinds of iterative control structures. One iterative loop control structure supports a fixed number of repetitions (e.g., beat the mixture 40 times; for each element of this list.). We traditionally call such structures **for loops**. Often, this control structure gives you an associated counter variable.

A second looping control structure allows you to repeat an action for as long as a condition holds (e.g., as long as the input has not all been read, read a line and process it; as long as the estimate is not close enough, refine the estimate). We tend to call such loops **while loops**. Unlike for loops, which typically call for a fixed number of repetitions, while loops typically have an unknown number of repetitions.

A third looping control structure is a slight variant of the while loop. In a **repeat loop** (Also referred as **Do-while** loops), work continues until a condition holds. For example, you might keep working on an assignment until it is correct. Repeat loops are quite similar to while loops, except that while loops check the condition before they execute the body of the loop and repeat loops check the condition afterwards.

Exercise

1. Make a class *Factorial* which consists of a public method factorial( ). Factorial method must take integer as an input and calculate its factorial. Class should be properly initialized with a constructor.

Hint:

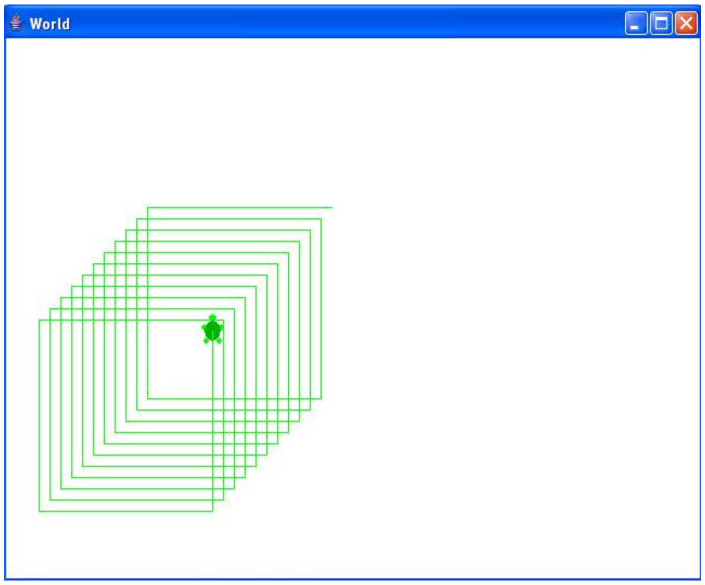
```
public class Factorial
   {
     int number;

// Constructor

  public Factorial(int n)
   {
    number=n;
    }

public void factorial(int number) {

 // Write your executable code here.

   }
public static void main(String args[])
  {
     Factorial fact =new Factorial( 6 );
     fact.factorial();

    }
}
```

2. Write a program using *while loop* that initializes a double variable and displays the next 10 numbers mentioning whether the numbers are even or odd. It should calculate and display the sum and average of the numbers.

Hint:

```
public class Average{

  double number;

  public Average(double n)
  {
    number=n;
  }

  public void evenodd()
  {
   // Write your executable code here.
  }

   public static void main(String args[])
  {
   Average average=new Average(3);  // Enter number here to initialize through constructor
   average.evenodd();
  }

}
```

3. Draw the following shape using **for loop**.



Hint:

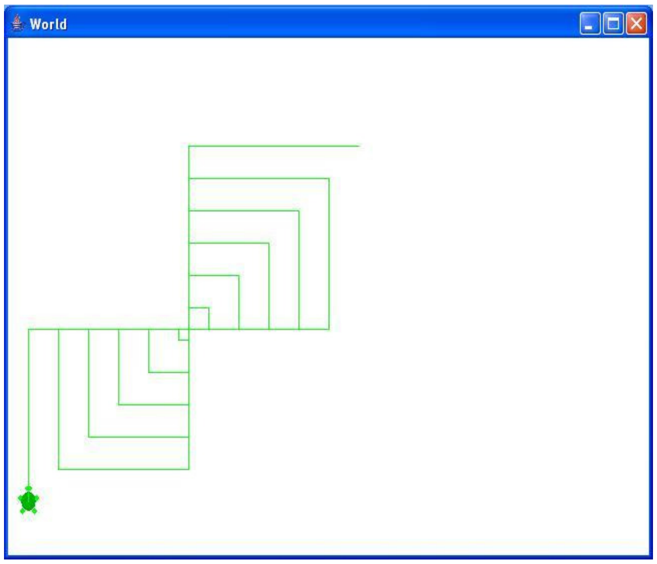
```
public class Loop{
     // start of class

  World world;
    Turtle turtle;

   public Loop()
    {
    world=new World();
   turtle=new Turtle(300,150,world);
    }
   public void spiralSquare()
    {
          // Write executable code here using for loop
     }

public static void main(String[] args)
 {
 Loop loopObj=new Loop();
 loopObj.spiralSquare();
 }

 }
```

4. Write a program that will display the factors of given number.

Hint:

```
              // Use while loop

          public class Factors
            {
             double number;

            public Factors(double n)
            {
              number=n;
             }

           public void factorCalc(){
            // Write your executable code here.
           }

             public static void main(String args[])
             {
              Factors fac=new Factors(9);
              fac.factorCalc();

              }
          }
```

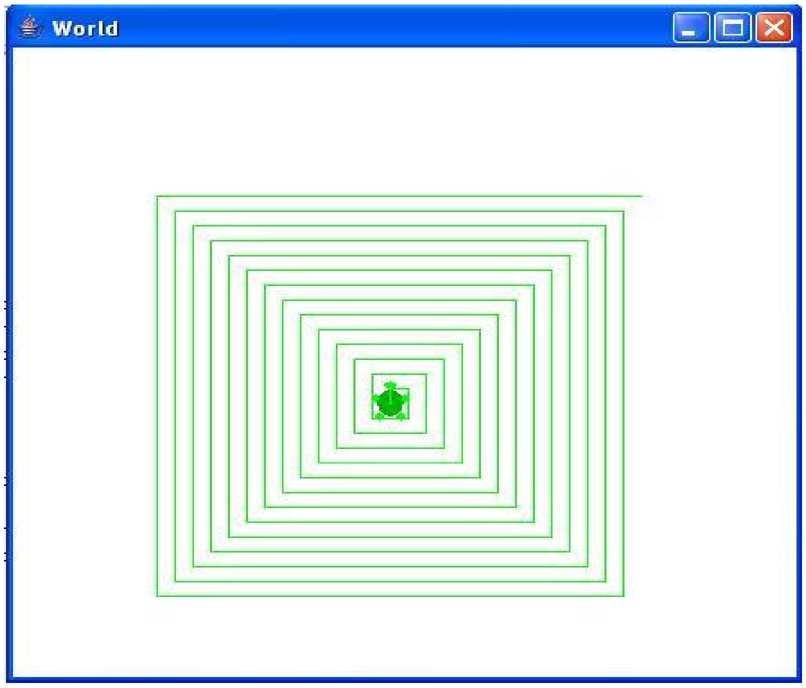
5. Draw the following figure.



Hint:

```
public class Loop{
// start of class

World world;
 Turtle turtle;

   public Loop()
    {
    world=new World();
   turtle=new Turtle(350,100,world);
    }
   public void spiralSquare()
    {
          // Write executable code here using for loop
     }

                public static void main(String[] args)
{
Loop loopObj=new Loop();
loopObj.spiralSquare();
}

}
```

6. Draw the following figure.



Hint:

```
public class Loop{
// start of class
World world;
 Turtle turtle;

 public Loop()
  {
  world=new World();
 turtle=new Turtle(350,100,world);
  }
 public void spiralSquare()
  {
       // Write executable code here using for loop
   }

              public static void main(String[] args)
{
Loop loopObj=new Loop();
loopObj.spiralSquare();
}

}
```