

Lab 6

For 1 through 7, use **both** a for loop and a while loop. For the rest of the questions, choose whether to use a for or while loop.

1. Print 50 to 10.

```
class Print50To10
{
    public static void main(String[] args)
    {
        for (int i = 50; i >= 10; i--)
        {
            System.out.println(i);
        }

        int i2 = 50;
        while (i2 >= 10)
        {
            System.out.println(i2);
            i2--;
        }
    }
}
```

2. Print 0 to 100. The output is kind of hard to read because it's so long. Even if you printed one per line or all on one line. Try to print 0 to 100 with 10 numbers per line. Hint: try using modulus %

```
System.out.println("\n\nQuestion 2:");
//for loop
for(int i = 0; i<=100; i++) {
    if(i % 10 == 0 && i != 0) {
        System.out.println();
    }
    System.out.print(i + " ");
}
System.out.println();

//while loop
i2 = 0;
while(i2 <= 100) {
    if(i2 % 10 == 0 && i2 != 0) {
        System.out.println();
    }
    System.out.print(i2 + " ");
}
```

```
        i2++;  
    }  
}
```

```
class Print0To100  
{  
    public static void main(String[] args)  
    {  
        for (int i = 0; i <= 100; i++)  
        {  
            System.out.print(i + " ");  
            if (i % 9 == 0 && i != 0)  
            {  
                System.out.println();  
            }  
        }  
  
        System.out.println();  
        int i2 = 0;  
        while (i2 <= 100)  
        {  
            System.out.print(i2 + " ");  
            if (i2 % 9 == 0 && i2 != 0)  
            {  
                System.out.println();  
            }  
            i2++;  
        }  
    }  
}
```

3. Print the alphabet 'A' to 'Z'.

- Hint: You can use char instead of int in your initialization statement.

```
class PrintAlphabet  
{  
    public static void main(String[] args)  
    {  
        for (char i = 'A'; i <= 'Z'; i++)  
        {  
            System.out.print(i + " ");  
        }  
  
        System.out.println();  
        char i2 = 'A';  
        while (i2 <= 'Z')  
        {  
            System.out.print(i2 + " ");  
        }  
    }  
}
```

```

        i2++;
    }
}

```

4. Print the alphabet backwards 'z' to 'a'.

```

class PrintAlphabetBackwards
{
    public static void main(String[] args)
    {
        for (char i = 'Z'; i >= 'A'; i--)
        {
            System.out.print(i + " ");
        }

        System.out.println();
        char i2 = 'Z';
        while (i2 >= 'A')
        {
            System.out.print(i2 + " ");
            i2--;
        }
    }
}

```

5. Print the sum of the even, positive integers less than 50.

```

class SumOfEven
{
    public static void main(String[] args)
    {
        //for loop
        int sum = 0;
        for (int i = 0; i < 50; i += 2)
        {
            sum += i;
        }
        System.out.println("Sum of even = " + sum);

        //while loop
        int sum2 = 0;
        int i2 = 0;
        while (i2 < 50)
        {
            sum2 += i2;
            i2 += 2;
        }
    }
}

```

```

    }
    System.out.println("Sum of even = " + sum2);
}
}

```

6. Count the numbers divisible by 2 or 7 between 20 to 300 inclusive.

- Remember: OR in Java is ||, AND is &&

```

class CountDivisible
{
    public static void main(String[] args)
    {
        //for loop
        int count = 0;
        for (int i = 20; i <= 300; i++)
        {
            if (i % 2 == 0 || i % 7 == 0)
            {
                count++;
            }
        }
        System.out.println("Count = " + count);

        //while loop
        int count2 = 0;
        int i2 = 20;
        while (i2 <= 300)
        {
            if (i2 % 2 == 0 || i2 % 7 == 0)
            {
                count2++;
            }
            i2++;
        }
        System.out.println("Count = " + count2);
    }
}

```

7. Count the number of odd numbers between 15 and 75 inclusive.

```

class CountOdd
{
    public static void main(String[] args)
    {
        //for loop
        int count = 0;

```

```

    for (int i = 15; i <= 75; i++)
    {
        if (i % 2 != 0)
        {
            count++;
        }
    }
    System.out.println("Count = " + count);

    //while loop
    int count2 = 0;
    int i2 = 15;
    while (i2 <= 75)
    {
        if (i2 % 2 != 0)
        {
            count2++;
        }
        i2++;
    }
    System.out.println("Count = " + count2);
}
}

```

8. Write a program that displays the following table. Kilograms should start from 1 to 199 and be odd.

- 1 kilogram is 2.2 pounds

Kilograms	Pounds
1	2.2
3	6.6
...	
197	433.4
199	437.8

```

class KilogramsToPounds
{
    public static void main(String[] args)
    {
        System.out.println("Kilograms    Pounds");
        for (int i = 1; i <= 199; i += 2)
        {
            System.out.println(i + "          " + i * 2.2);
        }
    }
}

```

9. Write a program to compute the sum of digits of any length integer.

Use Scanner to obtain the integer from the user.

```
```java
import java.util.Scanner;

class SumOfDigits
{
 public static void main(String[] args)
 {
 Scanner input = new Scanner(System.in);
 System.out.print("Enter an integer: ");
 int num = input.nextInt();
 int sum = 0;
 while (num != 0)
 {
 sum += num % 10;
 num /= 10;
 }
 System.out.println("Sum of digits = " + sum);
 }
}
```

10. Remember our factorial question from lab 2?

Factorial represented in mathematics by the symbol ! is the product of 1 to n. For example:

$$5! = 1 \times 2 \times 3 \times 4 \times 5 = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

Use a loop to make a program compute  $n! = 123 \times \dots \times n$ . Use Scanner to obtain n from a user.

```
import java.util.Scanner;

class Factorial
{
 public static void main(String[] args)
 {
 Scanner input = new Scanner(System.in);
 System.out.print("Enter an integer: ");
 int n = input.nextInt();
 int fact = 1;
 for (int i = 1; i <= n; i++)
 {
 fact *= i;
 }
 System.out.println("Factorial of " + n + " = " + fact);
 }
}
```

11. One way pi can be approximated is by the following summation:

$$\text{Pi} = 4 * (1 - 1/3 + 1/5 - 1/7 + 1/9 - 1/11 + \dots)$$

Write a program to approximate pi using the first 20 terms of the summation above.

Test your program with 20 terms, 200 terms, 2000 terms, 20000 terms.

Pi= 3.14159265359

- Notice that the more terms you sum the more accurate the value estimates pi. This is a tricky question. Think about how you can go about going back and forth to adding and subtraction every loop.

```
class ApproximatePi
{
 public static void main(String[] args)
 {
 double pi = 0;
 int denom = 1;
 for (int i = 0; i < 20000; i++)
 {
 if (i % 2 == 0)
 {
 pi += (1.0 / denom);
 }
 else
 {
 pi -= (1.0 / denom);
 }
 denom += 2;
 }
 pi *= 4;
 System.out.println("PI = " + pi);
 }
}
```