Week 2

1. **(Math: pentagonal numbers)** A pentagonal number is defined as n(3n−1)/2 for . . ., and so on. Therefore, the first few numbers are 1, 5, 12, 22, ... Write a method with the following header that returns a pentagonal number:

   ```
   public static int getPentagonalNumber(int n)
   ```

   Write a test program that uses this method to display the first 100 pentagonal numbers with 10 numbers on each line.

2. **(Display patterns)** Write a method to display a pattern as follows:

   ```
   1
   2 1
   3 2 1
   ...
   n n-1 ... 3 2 1
   ```

3. The method header is:

   ```
   public static void displayPattern(int n)
   ```

4. **(Conversions between Celsius and Fahrenheit)** Write a class that contains the following two methods:

   ```
   /** Convert from Celsius to Fahrenheit */
   public static double celsiusToFahrenheit(double celsius)
   /** Convert from Fahrenheit to Celsius */
   public static double fahrenheitToCelsius(double
   fahrenheit)
   ```

   The formula for the conversion is:

fahrenheit = (9.0 / 5) * celsius + 32

celsius = (5.0 / 9) * (fahrenheit – 32)

Write a test program that invokes these methods to display the following tables:

5. **(Reverse the numbers entered)** Write a program that reads ten integers and displays them in the reverse of the order in which they were read.
6. **(Average an array)** Write two overloaded methods that return the average of an array with the following headers:

```
public static int average(int[] array)

public static double average(double[] array)
```

Write a test program that prompts the user to enter ten double values, invokes this method, and displays the average value.

7. **(Strictly identical arrays)** The arrays list1 and list2 are strictly identical if their corresponding elements are equal. Write a method that returns true if list1 and list2 are strictly identical, using the following header:

```
public static boolean equals(int[] list1, int[] list2)
```

Write a test program that prompts the user to enter two lists of integers and displays whether the two are strictly identical. Here are the sample runs. Note that the first number in the input indicates the number of the elements in the list.

```
Enter list1: 5 2 5 6 1 6 Enter
Enter list2: 5 2 5 6 1 6 Enter
Two lists are strictly identical
```

8. **(Largest row and column)** Write a program that randomly fills in 0s and 1s into a 4-by-4 matrix, prints the matrix, and finds the first row and column with the most 1s. Here is a sample run of the program:

0011

0011

1101

1010

The largest row index: 2

The largest column index: 2

9. **(Student grade averages)** Create respective arrays for the following data:

	Grade 1	Grade 2	Grade 3	Grade 4	Grade 5
Jack	76	54	89	76	98
Robert	34	65	23	87	100
Edward	80	65	97	54	94
Eddie	63	75	33	75	87

Create a method which returns an array of integers that will hold the average of each student.

```
public static int[] calculateAverage(int[][] grades)
```

Create a method that prints all the arrays in a formatted way

```
public static void printStudents(String[] names, int[][]
grades, int[] average)
```

Create another method which sorts all the arrays according to the average

```
public static void sort(String[] names, int[][] grades,
int[] average)
```

Test all the methods:

- Calculate average
- Print the arrays
- Sort the arrays
- Print the arrays

10. **(Arguments: Min and Max)** Write two functions that get undetermined number of parameters to find the minimum or the maximum

```
public static int max(int... n)

public static int min(int... n)
```

Write a test program that displays the maximum and the minimum of a sequence of numbers.

11. **(Command Line Arguments: sum)** Sum the numbers taken as a command line arguments.
12. **Area and perimeter of a circle)** Write a program that displays the area and perimeter of a circle that has a radius of 5.5 using the following formula:

perimeter = 2 * radius * p

area = radius * radius * p

13. **(Print a table)** Write a program that displays the following table:

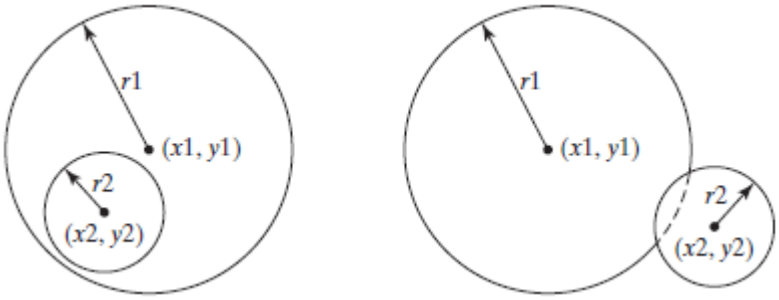
a	b	pow(a, b)
1	2	1
2	3	8
3	4	81
4	5	1024

14. **(Financial application: calculate interest)** If you know the balance and the annual percentage interest rate, you can compute the interest on the next monthly pay- ment using the following formula:

interest = balance * (annualInterestRate / 1200)

Write a program that reads the balance and the annual percentage interest rate and displays the interest for the next month.

15. **(Use input dialog)** Rewrite Financial application (1.3) using input and output dialog boxes.
16. **(Geometry: two circles)** Write a program that prompts the user to enter the center coordinates and radii of two circles and determines whether the second circle is inside the first or overlaps with the first, as shown in Figure:



(Hint: circle2 is inside circle1 if the distance between the two centers <= |r1 - r2| and circle2 overlaps circle1 if the distance between the two centers <= r1 + r2. Test your program to cover all cases.)

17. **(Display four patterns using loops)** Use nested loops that display the following patterns in four separate programs:

Pattern A	Pattern B	Pattern C	Pattern D
1	1 2 3 4 5 6	1	1 2 3 4 5 6
1 2	1 2 3 4 5	1 2	1 2 3 4 5
1 2 3	1 2 3 4	1 2 3	1 2 3 4
1 2 3 4	1 2 3	1 2 3 4	1 2 3
1 2 3 4 5	1 2	1 2 3 4 5	1 2
1 2 3 4 5 6	1	1 2 3 4 5 6	1

18. **(Compute e)** You can approximate e using the following series:

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \ldots + \frac{1}{i!}$$

Write a program that displays the e value for i = 10000, 20000, ..., and 100000.

19. **(Draw Full Diamond) Write a code with nested loops to display the following pattern:**

```
                1
              1 2 1
            1 2 3 2 1
          1 2 3 4 3 2 1
        1 2 3 4 5 4 3 2 1
      1 2 3 4 5 6 5 4 3 2 1
    1 2 3 4 5 6 7 6 5 4 3 2 1
  1 2 3 4 5 6 7 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1
  1 2 3 4 5 6 7 8 7 6 5 4 3 2 1
    1 2 3 4 5 6 7 6 5 4 3 2 1
      1 2 3 4 5 6 5 4 3 2 1
        1 2 3 4 5 4 3 2 1
          1 2 3 4 3 2 1
            1 2 3 2 1
              1 2 1
                1
```