Computer Science and Engineering Department
Object Oriented Concepts and Programming
JAVA Programming - Assignment 1 – Module1

1. Create a file called A.java, and in this file, declare a class called A. This class should define only one method called main(). In the body of this method, use System.out.println() to display the following pattern:

```
    A
   A A
  AAAAA
 A     A
A       A
```

2. Consider the following 3X3 matrix:

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

   - Write a Java program that first reads 9 doubles, representing the 9 letters above from the keyboard. It then outputs to the screen the determinant of the matrix.
   - Note: For a 3x3 matrix, the determinant is always equal to

$$a*(i*e-f*h)-b*(d*i-f*g)+\ c*\ (h*d-e*\ g)$$

3. Write a method named as ‘printTypeOfChar’. This method should take as input a char and print as follows. If the char is upper case, your method should print UPPERCASE. If the char is lower case, your method should print LOWERCASE. If the char is any other symbol, your method should print SYMBOL.

4. Write a method ‘countUppercase’. Your method should take as input a String and return an intrepresenting the number of upper case letters in the String. Now add a main method where you use the Scanner class to read a String from the user and call your method, outputting the returned result.

5. Write a program that generates a random number between 0 and 9 and then asks the user to guess a number between 0 and 9 and displays whether the guess is higher, lower or exactly the guessed number.

6. Write a program that generates a random integer within a range of two numbers. Write a method that takes a minimum integer and maximum integer from the user and returns a random integer between two input integers.

7. Write a program that generates a random integers in range of 150 to 200 until you generate the number ‘169’. When you generate the number ‘169’, print how many trials it took to do that.

8. Write a program to generate a pseudo-random password of 6 characters based on the following rules:
    - 1st character is a lowercase letter
    - 2nd character is a number 0-9
    - 3rd character is an uppercase letter
    - 4th character is a lowercase letter
    - 5th character is a number 0-9
    - 6th character is a number 0-9

   Note: all 6 characters should generate randomly.

   Example output is like: Generated password: h8Re62

9. Write a program to play Rock, Paper, Scissors. The outcome is as follows:

    - Rock crushes scissors -> Rock wins
    - Paper covers rock -> Paper wins
    - Scissors cuts paper -> Scissors win
    - If the same is chosen by player and computer, it's a draw.

   Initially the program will randomly generate a number of 0, 1, or 2 to represent Rock, Paper, and Scissors. The program then prompt the user to enter a 0, 1, or 2 and then display a message of who won.

10. A prime number is a positive number whose only even divisors are 1 and itself. Write a method isPrime that takes as input an integer n and returns a Boolean representing whether n is prime or not. Make sure to handle cases where the integer is negative. (Your method should return false in these cases).

11. Write a method firstPrimeNumbers which takes as input an int n and returns an int[]. The int[] should contain the first n prime numbers.

12. x is a factor of y if y is a multiple of x. Write a method calculateFactors. The method should take as input an int n and return an int[] containing all the factors of the number n.

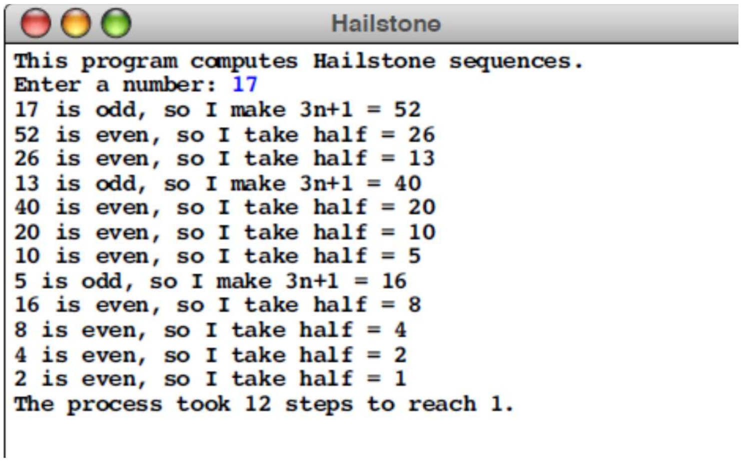
13. Write a method delete Element which takes as input an int[] and an int target and deletes all occurrences of target from the array. The method should return the new int[].

14. Write a program that finds whether input string is palindrome or not with the help of following methods:
    - The main method
    - A method that takes in a string and returns the reverse of that string
    - A method that checks whether or not a string is a palindrome

15. Write a program that reads in a number from the user and then displays the Hailstone sequence for that number. The Hailstone sequence can be expressed as follows:

    Pick some positive integer and call it *n*.
    If *n* is even, divide it by two.
    If *n* is odd, multiply it by three and add one.
    Continue this process until *n* is equal to one.

    The program output should be as follows:



```
This program computes Hailstone sequences.
Enter a number: 17
17 is odd, so I make 3n+1 = 52
52 is even, so I take half = 26
26 is even, so I take half = 13
13 is odd, so I make 3n+1 = 40
40 is even, so I take half = 20
20 is even, so I take half = 10
10 is even, so I take half = 5
5 is odd, so I make 3n+1 = 16
16 is even, so I take half = 8
8 is even, so I take half = 4
4 is even, so I take half = 2
2 is even, so I take half = 1
The process took 12 steps to reach 1.
```