LAB MANUAL
WEEK-2

Submission Deadlines in Teams Assignment: Q1, Q2, Q7
CSE-B & D: 19-08-2020 4.00PM
CSE-A & C: 21-08-2020 4.00PM
Submission Deadlines in Teams Assignment: Q3, Q4, Q5, Q6
CSE-B & D: 26-08-2020 4.00PM
CSE-A & C: 28-08-2020 4.00PM

1. Follow the Class Notes (Classes and Object) and complete the lab exercises(I to V) mentioned in the tutorial (Box Class Exercises).
2. Create a class **Person** with private member variable **name** of type *String*. Use getter and setter methods *public String getName( )* and *public setName(String name)*. Use a demo class with main ( ) to create an object of Person and using setName( ) and getName( ) to assign name to the member variable and print the value.

3. Create a class called **Complex** for performing arithmetic with complex numbers. Write a driver program to test your class.
   Complex numbers have the form realPart + imaginaryPart*i.
   Use floating-point variables to represent the private data of the class.
   - Provide a constructor method that enables an object of this class to be initialized when it is declared.
   - Provide a no argument constructor with default values in case no initializers are provided.
   - Provide public methods for each of the following:
     - Addition of two Complex numbers: The real parts are added together and the imaginary parts are added together.
     - Subtraction of two Complex numbers: The real part of the right operand is subtracted from the real part of the left operand and the imaginary part of the right operand is subtracted from the imaginary part of the left operand.
4. Write Java program involving two classes: OddAndEven and TestOddAndEven.
   OddAndEven has the following:
   - Instance variables countOfOdd and countOfEven both of type int

- A method addNumber that takes a number as parameter and increment countOfOdd, if the number is odd, else increment countOfEven.
- A method toString that returns a string in the form: "Number of Odd: x, Number of Even: y", where x and y are the values of the instance variables.

The TestOddAndEven class first creates OddAndEven object, then in a loop, read a number and use it to call the addNumber method until the user enters **Q**. Finally, it prints the count of odd and even numbers entered.

Java User Input Using Scanner

The Scanner class is used to get user input, and it is found in the java.util package. To use the Scanner class, create an object of the class

- To create an object of Scanner class, we usually pass the predefined object System.in, which represents the standard input stream. We may pass an object of class File if we want to read input from a file.
- To read numerical values of a certain data type XYZ, the function to use is nextXYZ(). For example, to read a value of type short, we can use nextShort()
- To read strings, we use nextLine().
- To read a single character, we use next().charAt(0). next() function returns the next token/word in the input as a string and charAt(0) function returns the first character in that string.

Method	Description
`nextBoolean()`	Reads a `boolean` value from the user
`nextByte()`	Reads a `byte` value from the user
`nextDouble()`	Reads a `double` value from the user
`nextFloat()`	Reads a `float` value from the user
`nextInt()`	Reads a `int` value from the user
`nextLine()`	Reads a `String` value from the user
`nextLong()`	Reads a `long` value from the user
`nextShort()`	Reads a `short` value from the user

Example to show scanner class working

```
// Java program to read data of various types using Scanner class.

import java.util.Scanner;
public class ScannerDemo1
{
	public static void main(String[] args)
	{
		// Declare the object and initialize with
		// predefined standard input object
		Scanner sc = new Scanner(System.in);

		// String input
		String name = sc.nextLine();

		// Character input
		char gender = sc.next().charAt(0);

		// Numerical data input
		// byte, short and float can be read
		// using similar-named functions.
		int age = sc.nextInt();
		long mobileNo = sc.nextLong();
		double cgpa = sc.nextDouble();

		// Print the values to check if the input was correctly obtained.
		System.out.println("Name: "+name);
		System.out.println("Gender: "+gender);
		System.out.println("Age: "+age);
		System.out.println("Mobile Number: "+mobileNo);
		System.out.println("CGPA: "+cgpa);
	}
```

Another example for scanner class

```
// Java program to read some values using Scanner
// class and print their mean.
import java.util.Scanner;

public class ScannerDemo2
{
	public static void main(String[] args)
	{
		// Declare an object and initialize with
		// predefined standard input object
		Scanner sc = new Scanner(System.in);

		// Initialize sum and count of input elements
		int sum = 0, count = 0;

		// Check if an int value is available
		while (sc.hasNextInt())
		{
			// Read an int value
			int num = sc.nextInt();
			sum += num;
			count++;
		}
		int mean = sum / count;
		System.out.println("Mean: " + mean);
	}
}
```

5. Write a class that represents a person's mailing address. It should have separate fields for the name, street address, city, state and ZIP code. Write an application AddressBook, to incorporate address book application :-
   (a) To add addresses.
   (b) Print addresses.

6. Implement a class Employee. An employee has a id(long int) , name (a string) , a salary (a double). Write a default constructor, a constructor with two parameters (name and salary), and getter methods to return the name and salary. Have a method raiseSalary (double byPercent) that raises the employee's salary by a certain percentage.

```
Employee harry = new Employee ("Hacker, Harry", 55000);
harry.raiseSalary(10); // Harry gets a 10% raise
```

7. The two cats (A & B) and a mouse(C) are at various positions on a line. Read their starting positions. Your task is to determine which cat will reach the mouse first, assuming the mouse doesn't move and the cats travel at equal speed. If the cats arrive at the same time, the mouse will be allowed to move and it will escape while they fight.

   Create a class **animal** with the member variables **cat_A**, **cat_B** and **mouse** (of type int). Implement a method **catAndMouse**( ).
   The variables **cat_A**, **cat_B** and **mouse** represent the positions of the two cats and mouse respectively. The **catAndMouse**( ) return the appropriate answer to each query, which will be printed on a new line.

   - If cat A catches the mouse first, *print Cat A*.
   - If cat B catches the mouse first, *print Cat B*.
   - If both cats reach the mouse at the same time, *print Mouse C* as the two cats fight and mouse escapes.

   Use **catMouseDemo** class with main function to implement the above scenario in main function by creating the object of class Animal.
   For example, cat A is at position 2 and cat B is at 5. If mouse C is at position 4, it is 2 units from cat A and 1 unit from cat B. Cat B will catch the mouse.
   Input the three positions and print output of catAndMouse( ).