

Lab 2 (100 pts)

Objectives:

- Learn to use class vs object methods
- Learn to use the methods of Java API classes, String, Math, Random and Scanner.

In this lab, you will work on a number of exercises illustrating the concepts of class vs object properties and library classes (String, Math, Random and Scanner).

Demos: Using Eclipse, create a project called **lab2_demos**. Copy the file, **ScannerDemo.java** underneath this folder. In this program, you will learn how to use a **Scanner** class in Java, to read keyboard input (and in later weeks, input from a file). The Scanner class is a class in the package, *java.util*, which allows the user to read values of various types. We will use this class, for now, to read numbers (integers and reals). In later weeks, you will use the other methods in the class.

Compile and run the program. You will be prompted to enter input. Enter an integer of your choice on a prompt for an integer. Enter an integer of your choice on a prompt for a float. Enter a real number with a decimal point and digits after the decimal point of your choice on a prompt for a double.

Did the output correctly show the values you have entered?

Now, run the program again. Enter a floating point number (with a decimal and digits after decimal) of your choice on a prompt for an integer. Did the output show correctly?

You will use the class **Scanner** to read in user input in Exercise1.

Ref: <http://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html>

Exercises

- In Eclipse, create a project called **lab2_exercises**.
- All the exercises in this lab should be included in a package called, **edu.scu.coen160.lab2_exercises**.
- Copy the posted files, **Exercise1-Exercise4.java** and **Main.java** into your working folder.

You will complete the code segments in each of the files using the comments in the code files and instructions given for each of the exercises. The program file, **Main.java** contains the **main()** which contains the code to run each of the tester methods in the program files. The **main()** in **Main.java** contains five (commented out) method calls to the **tester()** of each exercise. To compile the program, uncomment the Exercise you are testing and compile the **Exercisenumber.java** and **Main.java**. Run the program.

Exercise 1 (10 + 10 + 10 pts)

In this exercise, you will use **modulo division (%)** and a **conditional statement** in Java to check if a given year is a leap year or not.

Copy the code in the Java file, **Exercise1.java**. Include the package name, **edu.scu.coen160.lab2**. In this file, there are two classes, **CheckThings** and **Exercise1**.

Complete the method, **isItLeapYear()** in **CheckThings** class.

You will write the test code in **Exercise1** and test your method. The **tester1()** method in this class has hard-coded test cases. You will test your code in **CheckThings()** class using this test data. In part b) of this exercise, you will use the **Scanner** class to allow the user to input test-data of his/her choice.

To compile the program, uncomment the line, **Exercise1.tester1()** in **main()** method in **Main.java**. Run the program.

Based on your output, answer the following questions.

- a) Show (of the years given), the leap years. -----
- b) Now, use the **Scanner** class you have used in the **ScannerDemo.java** to read the user input for years. In the **Exercise1.java**, in class **Exercise1**, write a static method called **tester2()**. In this method, write code to create an instance of a **Scanner** to read in the user's input (as integers) for years. Check if the year input is a leap year or not.
Test your **tester2()** by calling it from **main()** of **Main** class, in **Main.java**.
- c) The method, **isItALeapYear()** is written as a class-level (static) method. Rewrite the method as an instance method. Comment out the testing code you have in **tester ()** method and insert code to test your instance method. Use the same years to test your code. **(10 pts)**

Exercise 2 (15 + 15 pts)

In this exercise, you will use the Java class String operations to complete the code segment given. Refer to <http://docs.oracle.com/javase/6/docs/api/java/lang/String.html>, for String API.

Copy the code in the Java file, **Exercise2.java**. Include the package name, **edu.scu.coen160.lab2_exericcses**.

In this file, there are two classes, **StringExercise** and **Exercise2**. You are required to complete the two methods, **fullName()** and **palindrome()** in StringExercise class.

You will write the test code in **Exercise2** and test your methods.

To compile the program, uncomment the line, **Exercise2.testter()** in **main()** method in **Main.java**. Run the program.

fullName(): Complete the method using the comments given. For a reference to String API, check the link given above.

palindrome():

A palindrome is a word, phrase or a sentence that reads the same when read backwards. The String class in Java does not have a method to reverse a string. There is another Java class called StringBuffer which has the method *reverse()*, that reverses a string. String objects in Java are immutable (cannot be changed), but **StringBuffer** objects are mutable.

Check the code given below to reverse a string using a **StringBuffer** object.

```
// original String
String strOriginal = "Hello World";

// create a StringBuffer object
StringBuffer tempStr = new StringBuffer(strOriginal);

// reverse the StringBuffer object
tempStr = tempStr.reverse();

// convert it back to a String, using toString()
String strReversed = tempStr.toString();

// print it
System.out.println("Reversed String : " + strReversed);
```

Use this approach to check if a given string is a palindrome or not. Complete the method, `palindrome()` using the comments given in the method. Test your method with the data given in the `tester()` in `Exercise2` class.

Points to note:

- a) When you are checking for a palindrome, ignore case. That is, Madam is a palindrome if you ignore case. Otherwise, it is not.
- b) When you are checking a sentence to see if it is a palindrome (“A man, a plan, a canal, Panama”, for example), you may have to consider just the text without commas and spaces (and ignore case as well).

Some of the String methods you may consider using (if and where applicable) are:

- a) `equals()`
- b) `charAt()`
- c) `substring()`
- d) `toLowerCase()`
- e) `toUpperCase()`
- f) `length()`

Please check the String API (link given above) and select the methods to use.

Random Numbers in Java

There are two ways to generate random numbers in Java, namely, using the methods in class `Random` or using the `random()` method in `Math` class.

- The [Random](#) class generates random integers, doubles, longs and so on, in various ranges.
- The static method `Math.random` generates *doubles* between 0 (inclusive) and 1 (exclusive).

Exercise 3 (15 pts)

In **Exercise3.java**, you will use the `random()` method in `Math` class.

Copy the code in the Java file, **Exercise3.java**. Include the package name, **edu.scu.coen160.lab2_exericises**. In this file, there are two classes, **RandomNumbers** and **Exercise3**.

Complete the method, **genRandomNum()** in class **RandomNumbers**, using the comments given. Test your code from the **tester()** in **Exercise3** class.

To compile the program, uncomment the line, **Exercise3.tester()** in **main()** method in **Main.java**. Run the program.

Exercise 4 (15+ 10 pts)

In **Exercise4.java**, you will use the `Random` class to generate the random numbers. Please refer to the API for `Random` given in the link below.

Copy the code in the Java file, **Exercise4.java**. Include the package name, **edu.scu.coen160.lab2_exericises**.

In this file, there are two classes, **RandomNumbers2** and **Exercise4**.

Complete the method, **genRandomNum()** in class **RandomNumbers2**, using the comments given. Test your code from the **tester()** in **Exercise4** class.

To compile the program, uncomment the line, **Exercise4.tester()** in **main()** method in **Main.java**. Run the program.

Ref: <http://docs.oracle.com/javase/6/docs/api/java/util/Random.html>

Based on your output, answer the following questions.

- a) For the `random()` in `Math` class (which generates doubles), what is the equivalent method in `Random` class? (5 pts)
- b) In `Random` class, when you specify a range as a parameter to `nextInt()` (`nextInt(10)`) for example, is 10 included in the range of numbers generated? (5 pts)