North South University

Department of Electrical and Computer Engineering

CSE 215L: Programming Language II Lab

Lab Manual - 2

Lab Instructor: Taif Al Musabe

Objective:

- To learn to basic mathematical and string methods
- To familiarize with **if-else** conditional statement

Mathematical Methods:

Method	Description
Math.abs()	It will return the Absolute value of the given value.
Math.max()	It returns the Largest of two values.
Math.min()	It is used to return the Smallest of two values.
Math.round()	It is used to round of the decimal numbers to the nearest value.
Math.sqrt()	It is used to return the square root of a number.
Math.cbrt()	It is used to return the cube root of a number.
Math.pow()	It returns the value of first argument raised to the power to second argument.
Math.signum()	It is used to find the sign of a given value.
Math.ceil()	It is used to find the smallest integer value that is greater than or equal to the argument or mathematical integer.
Math.copySign()	It is used to find the Absolute value of first argument along with sign specified in second argument.
Math.nextAfter()	It is used to return the floating-point number adjacent to the first argument in the direction of the second argument.
Math.nextUp()	It returns the floating-point value adjacent to d in the direction of positive infinity.
Math.nextDown()	It returns the floating-point value adjacent to d in the direction of negative infinity.
Math.floor()	It is used to find the largest integer value which is less than or equal to the argument and is equal to the mathematical integer of a double value.
Math.floorDiv()	It is used to find the largest integer value that is less than or equal to the algebraic quotient.
Math.random()	It returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.
Math.rint()	It returns the double value that is closest to the given argument and equal to mathematical integer.
Math.hypot()	It returns sqrt(x^2 +y^2) without intermediate overflow or underflow.
Math.ulp()	It returns the size of an ulp of the argument.
Math.getExponent()	It is used to return the unbiased exponent used in the representation of a value.
Math.IEEEremainder()	It is used to calculate the remainder operation on two arguments as prescribed by the IEEE 754 standard and returns value.
Math.addExact()	It is used to return the sum of its arguments, throwing an exception if the result overflows an int or long.
Math.subtractExact()	It returns the difference of the arguments, throwing an exception if the result overflows an int.
Math.multiplyExact()	It is used to return the product of the arguments, throwing an exception if the result overflows an int or long.
Math.incrementExact()	It returns the argument incremented by one, throwing an exception if the result overflows an int.
Math.decrementExact()	It is used to return the argument decremented by one, throwing an exception if the result overflows an int or long.
Math.negateExact()	It is used to return the negation of the argument, throwing an exception if the result overflows an int or long.
Math.toIntExact()	It returns the value of the long argument, throwing an exception if the value overflows an int.

String Methods:

Method	Description
char charAt(int index)	returns char value for the particular index
int length()	returns string length
static String format(String format, Object... args)	returns a formatted string.
static String format(Locale l, String format, Object... args)	returns formatted string with given locale.
String substring(int beginIndex)	returns substring for given begin index.
String substring(int beginIndex, int endIndex)	returns substring for given begin index and end index.
boolean contains(CharSequence s)	returns true or false after matching the sequence of char value.
static String join(CharSequence delimiter, CharSequence... elements)	returns a joined string.
static String join(CharSequence delimiter, Iterable<? extends CharSequence> elements)	returns a joined string.
boolean equals(Object another)	checks the equality of string with the given object.
boolean isEmpty()	checks if string is empty.
String concat(String str)	concatenates the specified string.
String replace(char old, char new)	replaces all occurrences of the specified char value.
String replace(CharSequence old, CharSequence new)	replaces all occurrences of the specified CharSequence.
static String equalsIgnoreCase(String another)	compares another string. It doesn't check case.
String[] split(String regex)	returns a split string matching regex.
String[] split(String regex, int limit)	returns a split string matching regex and limit.
String intern()	returns an interned string.
int indexOf(int ch)	returns the specified char value index.
int indexOf(int ch, int fromIndex)	returns the specified char value index starting with given index.
int indexOf(String substring)	returns the specified substring index.
int indexOf(String substring, int fromIndex)	returns the specified substring index starting with given index.
String toLowerCase()	returns a string in lowercase.
String toLowerCase(Locale l)	returns a string in lowercase using specified locale.
String toUpperCase()	returns a string in uppercase.
String toUpperCase(Locale l)	returns a string in uppercase using specified locale.
String trim()	removes beginning and ending spaces of this string.
static String valueOf(int value)	converts given type into string. It is an overloaded method.

The following program find the square root of the numbers and max between two numbers given by the programmer.

```
public class BasicMathematics{
   public static void main(String[] args){
      double x = 28;
      double y = 4;
      // return the square root of y
      System.out.println("Square root of y is: " + Math.sqrt(y));
      // return the maximum of two numbers
      System.out.println("Maximum number of x and y is: " +Math.max(x,
      y));
   }
}
```

The following program find the index of a character from string given by the programmer.

```
public class BasicString{
   public static void main(String[] args){
      String s1 = ="this is index of example";
      //passing substring
      int index1=s1.indexOf("is");
      //returns the index of is substring
      int index2=s1.indexOf("index");
      //returns the index of index substring
      System.out.println(index1+"  "+index2);//2 8
  }
}
```

Conditional Statement:

One-way if statement	Two-way if statement

`if(boolean-expression){`
`  statement(s);`
`}`
	`if (boolean-expression) {`
	`  statement(s)-for-the-true-case;`
	`} else`
	`{`
	`  statement(s)-for-the-false-case;`
	`}`

Nested if statement	**Multi-way if statement**

`if(Boolean-expression-1){`
`  if(Boolean-expression-2){`
`  statement(s);`
`  }`
`}`
	`if(Boolean-expression-1){`
	`  statement(s);`
	`}`
	`else if(Boolean-expression-2){`
	`  statement(s);`
	`} …`
	`…`
	`else if(Boolean-expression-n){`
	`  statement(s);`
	`} else{`
	`  statement(s);`
	`}`

Task – 1

(*Algebra: solve quadratic equations*) The two roots of a quadratic equation $ax^2 + bx + c = 0$ can be obtained using the following formula:

$$r_1 = \frac{-b+\sqrt{b^2-4ac}}{2a} \quad \textbf{and} \quad r_1 = \frac{-b-\sqrt{b^2-4ac}}{2a}$$

b^2-4ac is called the discriminant of the quadratic equation. If it is positive, the equation has two real roots. If it is zero, the equation has one root. If it is negative, the equation has no real roots. Write a program that prompts the user to enter values for *a*, *b*, and *c* and displays the result based on the discriminant. If the discriminant is positive, display two roots. If the discriminant is **0**, display one root. Otherwise, display "The equation has no real roots".

Task – 2

(*Palindrome number*) Write a program that prompts the user to enter a string and determines whether it is a palindrome. A string is palindrome if it reads the same from right to left and from left to right. Here is a sample run of this program:

```
Enter a string: racecar
racecar is a palindrome
```

```
Enter a string: john
john is not a palindrome
```

Task – 3

(*Random month*) Write a program that randomly generates an integer between 1 and 12 and displays the English month name January, February, …, December for the number 1, 2, …, 12, accordingly.

Homework – 1

(*Game: scissor, rock, paper*) Write a program that plays the popular scissor-rock-paper game. (A scissor can cut a paper, a rock can knock a scissor, and a paper can wrap a rock.) The program randomly generates a number **0**, **1**, or **2** representing scissor, rock, and paper. The program prompts the user to enter a number **0**, **1**, or **2** and displays a message indicating whether the user or the computer wins, loses, or draws. Here are sample runs:

```
scissor (0), rock (1), paper (2): 1
The computer is scissor. You are rock. You won
```

```
scissor (0), rock (1), paper (2): 2
The computer is paper. You are paper too. It is a draw
```

Homework – 2

(*Count positive and negative numbers and compute the average of numbers*) Write a program that reads an unspecified number of integers, determines how many positive and negative values have been read, and computes the total and average of the input values (not counting zeros). Your program ends with the input **0**. Display the average as a floating-point number. Here is a sample run:

```
Enter an integer, the input ends if it is 0: 1 2 -1 3 0
The number of positives is 3
The number of negatives is 1
The total is 5.0 The average is 1.25
```

Homework – 3

(*Find the vowel*)Write a program that reads string to check it contains any vowel or not. Here is a sample run:

```
Enter a string: fvcd
There is no vowel
Enter a string: aaaa
There are vowels
```