Assignment 1 – Wei Sun, May 4th

1. What is JDK? JRE? JVM?

JDK represents for **Java Development Kit**, which is a software development environment used for developing Java applications and applets. It includes the Java Runtime **Environment** (JRE), an interpreter/loader (Java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc), and other tools needed in Java development.

JRE stands for **Java Runtime Environment** and is also known as **"Java RTE"**. The Java Runtime Environment provides the minimum requirements for executing a Java application, it consists of Java Virtual Machine (JVM), core classes and supporting files.

JVM (**Java Virtual Machine**) is an engine that provides a runtime environment to drive the Java code or applications. It converts Java bytecode into machine language. JVM is part of JRE. It cannot be separately downloaded or installed. To install JVM, JRE installation is needed.

2. What is java compiler?

Java compiler is a compiler that converts a Java source code (.java file) into bytecode (.class file).

3. Why is java platform independent?

Java is platform-independent because it uses a virtual machine. The Java programming language and all APIs are compiled into bytecodes. Bytecodes are effectively platform-independent. The virtual machine takes care of the differences between the bytecodes for the different platforms. The run-time requirements for Java are therefore very small. The Java virtual machine takes care of all hardware-related issues, so that no code has to be compiled for different hardware.

4. What is IDE? Why is it important for developers?

IDE stands for **Integrated Development Environment**, which is a software suite that consolidates basic tools required to write and test software.

An IDE can improve the productivity of software developers thanks to fast setup and standardization across tools.Without an IDE, developers spend time deciding what tools to use for various tasks, configuring the tools and learning how to use them. Many or even all of the necessary dev-test tools are included in one integrated development environment.

5. Is java case sensitive?

Java is a case-sensitive language, which means that the upper or lower case of letters in your Java programs matter.

6. What do the following key words do?
   static, final, public, private, void, null, package, Class, new

   static: a non-access modifier used for methods and attributes. Static methods/attributes can be accessed without creating an object of a class

   final: The final keyword is a non-access modifier used for classes, attributes and methods, which makes them non-changeable (impossible to inherit or override)

   public: The public keyword is an access modifier used for classes, attributes, methods and constructors, making them accessible by any other class.

   private: The private keyword is an access modifier used for attributes, methods and constructors, making them only accessible within the declared class.

   void: The void keyword specifies that a method should not have a return value.

   null: In Java, null is a reserved word for literal values. It seems like a keyword, but, it is a literal similar to true and false.

   package: The package keyword creates a package.

   class: The class keyword is used to create a class.

   new: The new keyword creates new objects.

7. What is primitive type and reference type?

   Types in Java are divided into two categories—primitive types and reference types.

   The primitive types are boolean, byte, char, short, int, long, float and double.

   All other types are reference types, so classes, which specify the types of objects, are reference types.

8. Is parameter passed by value or reference?

   All object references in Java are passed by value. This means that a copy of the value will be passed to a method. But the trick is that passing a copy of the value also changes the real value of the object.

9. What is the output: System.out.println(1 > 0 ? “A”:”B”);

   The output would be “A”. This is ternary operation in Java, the syntax is:

   Expression1 ? Expression2 : Expression3

   The Expression1 should be a Boolean expression, if it is true, this operation will return Expression2, else, it will return Expression3

10. How to define constants in java?

    Use the keyword “final” to turn the variable into constants.

11. What is String? Is it primitive type?

    A string is traditionally a sequence of characters. It is not a primitive type, instead it is an object.

12. How to check if a String is representing a number?

Users tend to mistype input values fairly often, which is why developers have to hold their hand as much as possible during IO operations.

The easiest way of checking if a String is a numeric or not is by using one of the following built-in Java methods:

Integer.parseInt()

Integer.valueOf()

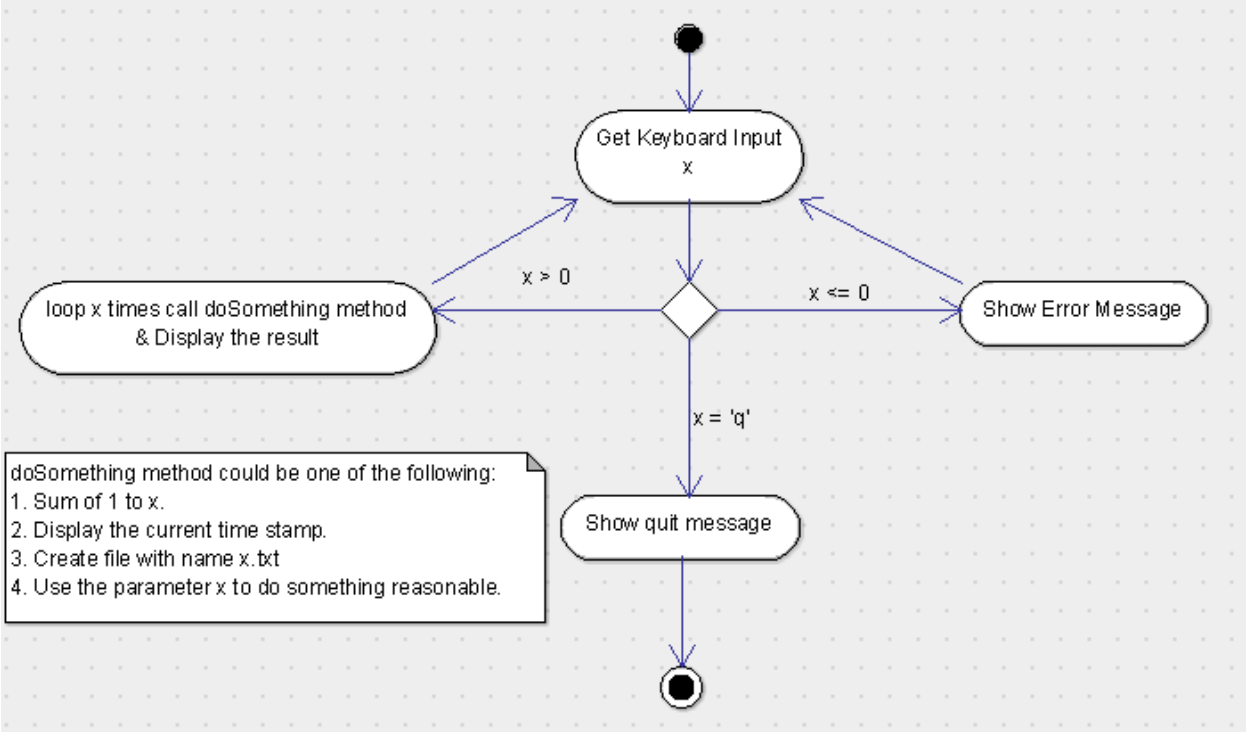
Double.parseDouble()

Float.parseFloat()

Long.parseLong()

These methods convert a given String into its numeric equivalent. If they can't convert it, a *NumberFormatException* is thrown, indicating that the String wasn't numeric.

13. Write a program to implement the following activity diagram:



```
public class solution {

    private static String quitMessage = "Successfully quited!";
    private static String errorMessage = "Error!";

    public static void main(String[] args) {
        Scanner in = new Scanner(new BufferedReader(new
InputStreamReader(System.in)));
        String x = in.next();
        while(!x.equals("q")) {
            try {
                int val = Integer.parseInt(x);
                if(val <= 0) {
                    System.out.println(errorMessage);
                    break;
                } else {
                    int sum = 0;
                    for(int i = 1; i <= x; i++) {
                        sum += i;
                    }
                    System.out.println(x);
                }
            } catch(NumberFormatException e) {
                System.out.println(errorMessage);
            }
            x = in.next();
        }
        System.out.println(quitMessage);
    }
}
```

14. Write a program to merge two array of int.

```
private int[] mergeArrays(int[] a, int[] b) {
    int len1 = a.length, len2 = b.length;
    int[] res = new int[len1+len2];

    for(int i = 0; i < len1 + len2; i++) {
        if(i >= len1) {
            res[i] = b[i-len1];
        } else {
            res[i] = a[i];
        }
    }

    return res;
}
```

15. Write a program to find the second largest number inside an array of int.

```
private int secondLargest(int[] arr) {
    if(arr.length < 2)
        return null;
    int largest = Integer.MIN_VALUE, secondLargest = Integer.MIN_VALUE;

    for(int ele : arr) {
        if(ele >= largest) {
            secondLargest = largest;
            largest = ele;
        } else if (ele > secondLargest) {
            secondLargest = ele;
        }
    }

    return secondLargest;
}
```