Assignment 1

1. What is JDK? JRE? JVM?

JDK is a software development kit whereas JRE is a software bundle that allows Java program to run, whereas JVM is an environment for executing bytecode.

2. What is java compiler?

A Java compiler is a program that takes the text file work of a developer and compiles it into a platform-independent Java 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

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

An integrated development environment (IDE) is a software suite that consolidates basic tools required to write and test software.

An IDE, or Integrated Development Environment, enables programmers to consolidate the different aspects of writing a computer program. IDEs increase programmer productivity by combining common activities of writing software into a single application: editing source code, building executables, and debugging.

5. Is java case sensitive?

Java is a case-sensitive language.

6. What do the following key words do?

static: The static keyword in Java is used to share the same variable or method of a given class.

final: In Java we use final keyword with variables to specify its values are not to be changed.

public:public is a Java keyword which declares a member's access as public. Public members are visible to all other classes.

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

package: Package in Java is a mechanism to encapsulate a group of classes, sub packages and interfaces.

class: A Class is like an object constructor, or a "blueprint" for creating objects.

new: new keyword is used to create an instance of the class.

7. What is primitive type and reference type?

Primitive types are the basic types of data: byte, short, int, long, float, double, boolean, char. Primitive variables store primitive values. Reference types are any instantiable class as well as arrays: String, Scanner, Random, Die, int[ ], String[ ]

8. Is parameter passed by value or reference?

By value

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

A

10. How to define constants in java?

We cloud use keyword “final”.

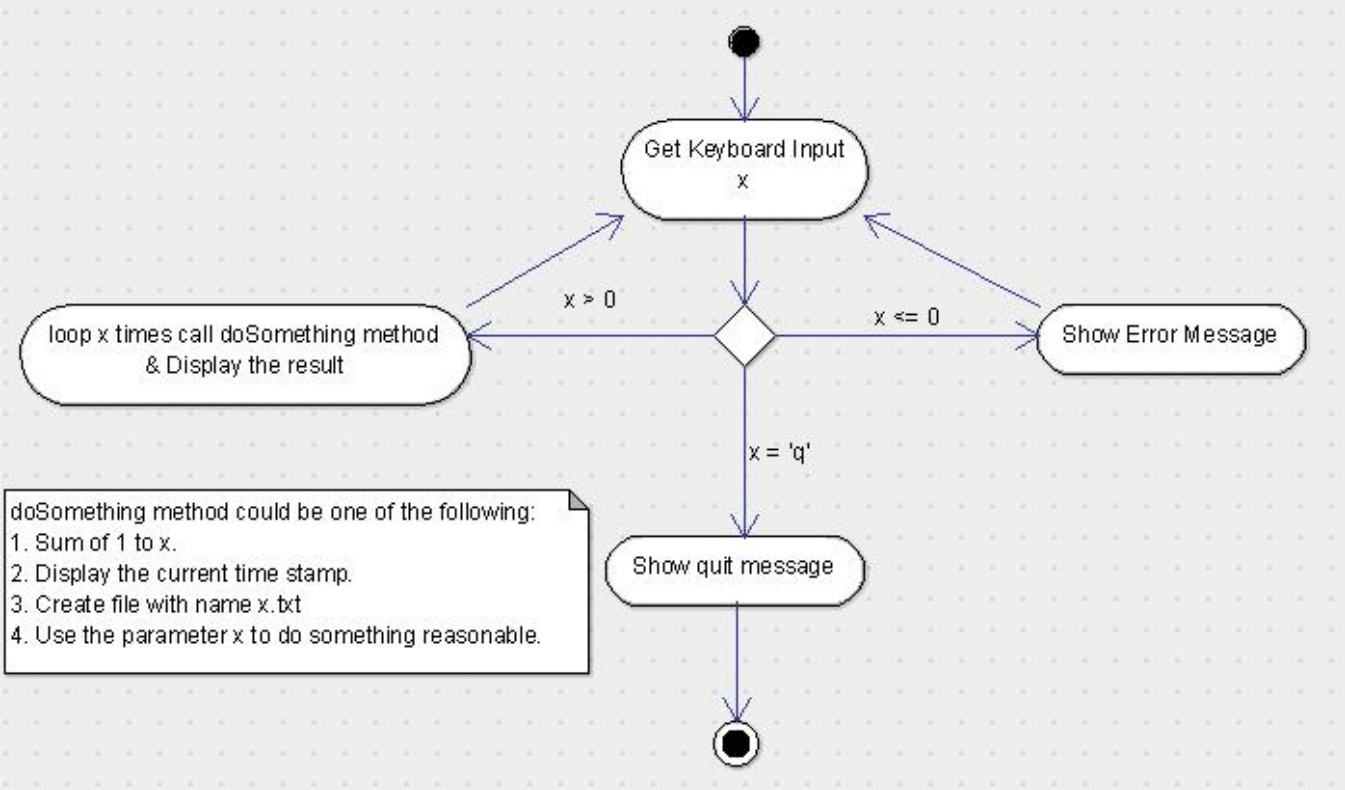
11. What is String? Is it primitive type?

String is basically an object that represents sequence of char values. And string is not primitive type.

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

We could use Integer.valueOf( ) method to convert string. If it dose work, this string is representing a number.

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




```
System.out.println(“Please input number:);
Scanner reader = new Scanner(System.in);
While(reader.nextLine( ) !=’q’){
        if(reader.nextInt( ) <= 0){
                System.out.println(“Wrong number, please input new number!)
                reader = new Scanner(System.in);
        }else{

                int num= reader.nextInt( )
                int sum = 0;
                for( int i=1; i < = num;i++){

                        sum=sum+i;
                        System.out.println(“sum is equal to “+sum);
```

```
                }
                DateTimeFormatter dtf   =DateTimeFormatter.ofPattern("yyyy/MM/dd
        HH:mm:ss");
                LocalDateTime now = LocalDateTime.now();
                System.out.println(dtf.format(now));

                File mytxt = new File("x.txt");
                mytxt.createNewFile();
                FileWriter myWriter = new FileWriter("x.txt");
                myWriter.write("x");
                myWriter.close();
        }
        System.out.println(“Please input new number:);
        reader = new Scanner(System.in);
}
```

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

```
    public int[] merge(int[] arr1,int [] arr2) {
                int len1=arr1.length;
                int len2=arr2.length;
                int len = len1+len2;

                int[] newarr =new int[len];
                for(int i=0;i<len1;i++) {
                        newarr[i] = arr1[i];
                }
                for(int i=0;i<len2;i++) {
                        newarr[i+len1]=arr2[i];
                }
                return newarr;

        }
```

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

```
    public int findSecondLargest(int[] nums) {
        PriorityQueue<Integer> list = new PriorityQueue<Integer>((n1,n2)->n1-n2);
        for(int num:nums){
            list.add(num);
            if(list.size()>2){
                list.poll();
            }
        }
        Integer tem=list.poll();
        if(tem==null) return -1;
        else return tem;
    }
```