Assignment 2

1. Why we need packages in java?
   We need packages to prevent naming conflicts.

2. What is the default imported package?
   The java.lang package.

3. What is Class? What is Object?
   A class is a non-primitive or user-defined data type in Java, while an object is an instance of a class.

4. Why we need constructor?
   We need constructors to initialize the state inside an object.

5. What is the default value of local variable? What is the default value of instance variable?
   There is no default value for local variables.
   The default value of instance variable is 0.

6. What is garbage collection?
   Garbage collection is a process which JVM deletes the variables that will never be accessed on heap.

7. The protected data can be accessed by subclasses or same package. True or false?
   True.

8. What is immutable class?
   If a class is immutable, it's state cannot be changed once it has been created.

9. What's the difference between "==" and equals method?
   For primitive types, they only use == and there is no equals method for them.
   For reference types, == always compares their class instance. If they refer to the same object, true; otherwise, false. And equals method is the same as == by default, however, the developer can override it to compare other things.

10. What is wrapper class?
    Wrapper classes provide a way to use primitive types as objects.
    Primitive type: byte, short, int, long, float, double, char, boolean
    Wrapper class: Byte, Short, Integer, Long, Float, Double, Character, Boolean

11. What is autoboxing?
    Autoboxing is the automatic conversion from primitive types to their wrapper classes made by Java compiler.
    If the conversion is from wrapper classes to their primitive types, unboxing.

12. StringBuilder is threadsafe but slower than StringBuffer, true or false?
    False. StringBuilder is not threadsafe but faster than StringBuffer.

13. Constructor can be inherited, true or false?
    False. Only members are inherited but constructors are not members.

14. How to call a super class's constructor?
    We can use super() to explicitly call the superclass constructor from subclass constructor. And super() must be the first statement inside the subclass constructor.

15. Which class is the super class of all classes?
    Object class.

16. Create a program to count how many files/folders are there inside one folder.
    - the count method should take a parameter called Criteria like this: count(Criteria criteria){}
    - For Criteria class, multiple conditions should be included such as: folder path, includeSubFolder or not, the extension of the file be counted and so on.
    - Optional: Take the input from keyboard.
    - Take care of the invalid inputs. Exception handling.
    - Get proper result displayed.
      "There are XXX file(s) and XXX folder(s) inside folder XXX with extension XXX." or something user friendly.

```
package Day2.HomeWork;


import java.io.File;

public class CountFiles {
    public static void main(String[] args) {
        Criteria c = new Criteria("/Users/lys/Documents/Antra");
        count(c);
    }
    public static void count(Criteria c) {
        int[] result = new int[]{0, 0};
        String path = new String(c.folderPath);
        File file = new File(path);
        countFileFolder(c, result);
        System.out.print("There are " + result[0] + " file(s) and " +
result[1] + " folder(s) ");
        System.out.print("inside folder " + file.getName() + ".");
    }

    // num[0] - number of files in this directory, num[1] - number of folders
in this directory
    public static void countFileFolder(Criteria c, int[] num) {
        if (c.file.isFile()) { // the path is a file's path
```

```
            num[0]++;
            return;
        }
        if (!c.includeSubFolder) { // no more subfolders
            num[0] += c.file.list().length;
            return;
        }

        // valid folder and there are subfolders
        for (File f : c.file.listFiles()) {
            if (f.isFile()) {
                num[0]++;
                //System.out.println("File: " + f.getName());
            }
            if (f.isDirectory()) {
                num[1]++;
                //System.out.println("Folder: " + f.getName());
                countFileFolder(new Criteria(f.getPath()), num);
            }
            // System.out.println(f.getName());
        }
    }
}
```

```
package Day2.HomeWork;

import java.io.File;

public class Criteria {
    static String folderPath;
    static boolean includeSubFolder;
    static String extension;
    static File file;

    Criteria(String path) {
        this.folderPath = path;
        this.file = new File(folderPath);
        this.includeSubFolder = isIncludeSubFolder();
    }

    public static boolean isFolder() {
        return file.isDirectory();
    }

    public static boolean isIncludeSubFolder() {
        if (!isFolder()) { // not a path of folder
            return false;
        }
        for (File f : file.listFiles()) {
            if (f.isFile()) {
```

```
                return true;
            }
        }
        return false;
    }
}
```