Assignment 2

1. Why we need packages in java?
   Prevent naming conflicts, control access, make searching, locating, usage of classes
2. What is the default imported package?
   Java.lang package is default imported by compiler
3. What is Class? What is Object?
   Class describes the contents of the objects that belong to it: like fields and methods
   Object is an instance of a class
4. Why we need constructor?
   Constructor can initialize the fields of an object
5. What is the default value of local variable? What is the default value of instance variable?
   Local variable has no default values
   Instance variable:
      primitive type: boolean -> false, numbers -> 0
      Reference type: null
6. What is garbage collection?
   Java program run automatically in order to manage memory space
7. The protected data can be accessed by subclasses or same package. True or false?
   True
8. What is immutable class?
   Once the object is created, we can't change its content anymore
   For example : All wrapper class and String class are all immutable class
9. What's the difference between "==" and equals method?
   "=="compares the reference addresses between two objects
   Equals method compares the content of two objects
10. What is wrapper class?
    A class that its objects wrap primitive types data
11. What is autoboxing?
    A conversion from primitive type data to its wrapper class without manually operation.
12. StringBuilder is threadsafe but slower than StringBuffer, true or false?
    False, StringBuffer is thread safe and slower than StringBuilder
13. Constructor can be inherited, true or false?
    True
14. How to call a super class's constructor?
    Using "super()" method
15. Which class is the super class of all classes?
    Object class is the super class of all classes
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.

```
import java.io.File;
import java.io.FileNotFoundException;

public class CountFile {
    public static void main(String[] args) throws FileNotFoundException {
        CountFile c = new CountFile();
        Criteria cr = new Criteria();
        cr.path = "D:\\Tomcat\\apache-tomcat-9.0.46-windows-x86\\apache-tomcat-9.0.46";
        File f = new File(cr.path);
        if (!f.exists()) {
            throw new FileNotFoundException("Can't Find this file");
        } else if (f.isDirectory()) {
            cr.include = true;
        } else {
            cr.include = false;
        }
        cr.extension = "no";
        c.count(cr);
    }
    public void count(Criteria c) throws FileNotFoundException {
        File f = new File(c.path);
        if (!f.exists()) {
            throw new FileNotFoundException("Can't Find this file");
        }
        int files = 0;
        int folders = 0;
        if (!c.include) {
            System.out.println("There are " + files + " file(s) and " + folders + " folder(s) inside
folder " + f.getName() + " with " + c.extension + " extension.");
            return;
        } else {
            File[] list = f.listFiles();
            for (int i = 0; i < list.length; i++) {
                if (list[i].isFile()) {
                    files++;
```

```
                } else {
                    folders++;
                }
            }
        }
        System.out.println("There are " + files + " file(s) and " + folders + " folder(s) inside folder "
+ f.getName() + " with " + c.extension + " extension.");
    }
}
class Criteria {
    String path;
    boolean include;
    String extension;
    public Criteria() {
    }
    public Criteria(String path, boolean include, String extension) {
        this.path = path;
        this.include = include;
        this.extension = extension;
    }
}
```