Assignment 2

1. Why we need packages in java?
2. What is the default imported package?
3. What is Class? What is Object?
4. Why we need constructor?
5. What is the default value of local variable? What is the default value of instance variable?
6. What is garbage collection?
7. The protected data can be accessed by subclasses or same package. True or false?
8. What is immutable class?
9. What's the difference between "==" and equals method?
10. What is wrapper class?
11. What is autoboxing?
12. StringBuilder is threadsafe but slower than StringBuffer, true or false?
13. Constructor can be inherited, true or false?
14. How to call a super class's constructor?
15. Which 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.

1. Why we need packages in java?
   Packages are used in Java to prevent naming conflicts, to control access, to make searching/locating and usage of classes, interfaces, enumerations, and annotations easier,
2. What is the default imported package?
   java.lang
3. What is Class? What is Object?
   Class is a blueprint for objects. A class is a user-defined type that describes what a certain type of object will look like. Object is an encapsulation of data along with functions that act upon that data.
4. Why we need constructor?
   The purpose of a Java constructor is to initializes the newly created object before it is used.
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, so local variables should be declared, and an initial value should be assigned before the first use.
   For numbers, the default value is 0, for Booleans it is false, and for object references it is null.
6. What is garbage collection?
   Garbage collection is the process by which Java programs perform automatic memory management. Java programs compile to bytecode that can be run on a Java Virtual Machine, or JVM for short. When Java programs run on the JVM, objects are created on the heap, which is a portion of memory dedicated to the program. Eventually, some objects will no longer be needed. The garbage collector finds these unused objects and deletes them to free up memory.
7. The protected data can be accessed by subclasses or same package. True or false?
   True.
8. What is immutable class?
   Immutable class in java means that once an object is created, we cannot change its content.
9. What's the difference between "==" and equals method?
   The main difference between the .equals() method and == operator is that one is a method, and the other is the operator. We can use == operators for reference comparison (address comparison) and .equals() method for content comparison.
10. What is wrapper class?
    A Wrapper class is a class whose object wraps or contains primitive data types. When we create an object to a wrapper class, it contains a field and in this field, we can store primitive data types.
11. What is autoboxing?
    Automatic conversion of primitive types to the object of their corresponding wrapper classes is known as autoboxing.
12. StringBuilder is threadsafe but slower than StringBuffer, true or false?
    False. StringBuffer is threadsafe but slower than StringBuilder.

13. Constructor can be inherited, true or false?
    False. It would not have much sense to inherit a constructor, since constructor of class A means creating an object of type A, and constructor of class B means creating an object of class B.
14. How to call a super class's constructor?
    super()
15. Which class is the super class of all classes?
    Object.
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.

```
public class Assignment2 {
    public static void main(String[] args) {
        Criteria criteria = new
Criteria("/Users/llm/Documents/Antra/assignments/AntraHW",true,".java");
        try {
            count(criteria);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    public static void count(Criteria criteria) throws IOException {
        String filePath = criteria.getFilePath();
        Path path = Paths.get(filePath);

        if (Files.isDirectory(path)) {

            Map<String, Long> result = Files.list(path).filter(f ->
f.toFile().isFile()).map(Assignment2::getExtension)
                    .collect(Collectors.groupingBy(Function.identity(),
Collectors.counting()));

            System.out.println(result);
        } else {
            System.out.printf("Path was not found.");
        }

    }

    public static String getExtension(Path path) {
        String parts[] = path.toString().split("\\.");
        if (1 < parts.length) {
            return parts[parts.length - 1];
        }
```

```
        return path.toString();
    }
}
class Criteria{
    private String filePath;
    private boolean includeSubFolder;
    private String extension;

    public Criteria(String filePath, boolean includeSubFolder, String
extension) {
        this.filePath = filePath;
        this.includeSubFolder = includeSubFolder;
        this.extension = extension;
    }

    public String getFilePath() {
        return filePath;
    }

    public void setFilePath(String filePath) {
        this.filePath = filePath;
    }

    public boolean isIncludeSubFolder() {
        return includeSubFolder;
    }

    public void setIncludeSubFolder(boolean includeSubFolder) {
        this.includeSubFolder = includeSubFolder;
    }

    public String getExtension() {
        return extension;
    }

    public void setExtension(String extension) {
        this.extension = extension;
    }
}
```