Assignment 2

1. Why we need packages in java?
   Package is the mechanism in java used to organize classes and prevent name collisions.

2. What is the default imported package?

   a. "java.lang" is the package which contains very common components.
   b. Second package is the "unnamed package", it contains all the components(classes,interfaces..etc).
   c. And third is obvious the "named package" which is the current package (the package in which the current file is defined)

3. What is Class? What is Object?
   A class is a user defined blueprint or prototype from which objects are created.
   An Object is the basic unit of OOP represents an entity.

4. Why we need constructor?
   Constructor provides a way to initialize an object and its member variables.

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.
   Instance variables have default values. For numbers, the default value is 0, for Booleans it is false, and for object references it is null. Values can be assigned during the declaration or within the constructor.

6. What is garbage collection?
   Automatic garbage collection is the process of looking at heap memory, identifying which objects are in use and which are not, and deleting the unused objects.

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

8. What is immutable class?
   Immutable class means that once an object is created, we cannot change its content.

9. What’s the difference between “==” and equals method?
   1. .equals() is method and == is operator

2. == operators for reference comparison (address comparison) and .equals() method for content comparison.

10. What is wrapper class?
When we create an object to a wrapper class, it contains a field which contains primitive data types. In other words, we can wrap a primitive value into a wrapper class object.

11. What is autoboxing?
Autoboxing is automatic conversion of primitive types to the object of their corresponding wrapper classes.

12. StringBuilder is thread safe but slower than StringBuffer, true or false?
False, StringBuilder is faster because it's not synchronized.

13. Constructor can be inherited, true or false?
True

14. How to call a super class's constructor?
Super();

15. Which class is the super class of all classes?
java.lang.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.

```java
//question2_16.java
package net.anra.pack1;
import java.io.File;
import java.util.Scanner;
public class question2_16 {
     /**
      * 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){}

      * 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.
      *
      * */

     private static void count(final Criteria c) {
          // String path, boolean includeSubFolder, String
extensionName
          int fileCount = 0;
          int folderCount = 0;
          int exFileCount = 0;
          try {
               final File folder = new File(c.path);
               File[] fileList = folder.listFiles();

               for (final File fileEntity : fileList) {
                    if (fileEntity.isDirectory()) {
                         folderCount++;
                         if (c.includeSubFolder) {
                              fileCount++;
                         }
                    } else {
                         fileCount++;
                    }
                    if
(fileEntity.getName().lastIndexOf(c.extensionName) != -1) {
                         // error if two ext name
                         exFileCount++;
                    }
               }
          } catch (Error e) {
```

```
                System.out.println(e);
            } finally {
                if (c.includeSubFolder) {
                    System.out.printf("There are %d file(s)
including %d folder(s) inside folder %s with extension %d.",
                            fileCount, folderCount, c.path,
exFileCount);
                } else {
                    System.out.printf("There are %d file(s) and
%d folder(s) inside folder %s with extension %d.",
                        fileCount, folderCount, c.path,
exFileCount);
                }
            }
        }

        public static void main(String[] args) {
            Scanner reader = new Scanner(System.in);  // Reading
from System.in
            System.out.println("enther path, includeSubFolder,
extension name");
            String path = reader.nextLine();// Scans the next
token of the input as an int.
            boolean includeSubFolder = reader.nextLine().charAt(0)
== 'y';
            String extensionName = reader.nextLine();

            //once finished
            reader.close();

            count(new Criteria(path, includeSubFolder, "." +
extensionName));

        }
}


// Criteria.java
package net.anra.pack1;

public class Criteria {
        String path;
        boolean includeSubFolder;
        String extensionName;
```

```
     public Criteria(String path, boolean includeSubFolder,
String extensionName) {
          /**
           * 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.
           * */
          this.path = path;
          this.includeSubFolder = includeSubFolder;
          this.extensionName = extensionName;

     }

}
```