Assignment 2 – Wei Sun, May 5th

1. Why we need packages in java?

A Java package is a mechanism for organizing Java classes into namespaces similar to the modules of Modula. Java packages can be stored in compressed files called JAR files, allowing classes to download faster as a group rather than one at a time. Programmers also typically use packages to organize classes belonging to the same category or providing similar functionality.

So, packages in Java are simply a mechanism used to organize classes and prevent class name collisions.

2. What is the default imported package?

Java compiler imports **java.lang package** internally by default. It provides the fundamental classes that are necessary to design a basic Java program.

3. What is Class? What is Object?

A **class** is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity.

An **object** is an instance of a class. A class is a template or blueprint from which objects are created. So, an object is the instance(result) of a class.

4. Why we need constructor?

A constructor is a special method of a class that initializes new objects or instances of the class.

5. What is the default value of local variable? What is the default value of instance variable?

The **local variables** do not have any default values in Java. This means that they can be declared and assigned a value before the variables are used for the first time, otherwise, the compiler throws an error.

The **instance variables** have default values, the values depend on the data types. 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?

Java garbage collection is the process by which Java programs perform automatic memory management. Java applications obtain objects in memory as needed. It is the task of garbage collection in the JVM to automatically determine what memory is no longer being used by a Java application and to recycle this memory for other uses.

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

True. Protected data member and method are only accessible by the classes of the **same package** and **the subclasses present in any package**.

8. What is immutable class?

Immutable class in java means that once an object is created, we cannot change its content. In Java, all the wrapper classes (like Integer, Boolean, Byte, Short) and String class is immutable.

9. What's the difference between "==" and equals method?

"==" operator is used for **reference comparison (address comparison)** and .equals() method is used for **content comparison**. In simple words, "==" checks if both objects point to the same memory location whereas .equals() evaluates to the comparison of values in the objects. If a class does not override the equals method, then by default, it uses the equals() method of **the closest parent class that has overridden this method**.

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?

Autoboxing is the automatic conversion that the Java compiler makes between the primitive types and their corresponding object wrapper classes. For example, converting an int to an Integer, a double to a Double, and so on. If the conversion goes the other way, this is called unboxing.

12. StringBuilder is threadsafe but slower than StringBuffer, true or false?

False. On the contrary, StringBuffer class is threadsafe, while StringBuilder is not. And as StringBuffer is threadsafe and synchronized, it is less efficient than StringBuilder.

13. Constructor can be inherited, true or false?

False. In Java, constructor in super class cannot be inherited by subclasses.

14. How to call a super class's constructor?

In Java, the superclass constructor can be called from **the first line of a subclass constructor** by using the special keyword **super()** and passing appropriate parameters.

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

Object class is the root class of the class hierarchy in Java.

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.

```
static class Criteria {
   public String folderPath;
   public boolean includeSubFolder;
   public String extension;

   public Criteria() {

   }

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

public static int[] count(Criteria criteria) {
   String folderPath = criteria.folderPath;
   String extension = criteria.extension;
   boolean includeSubFolder = criteria.includeSubFolder;
   try {
      File f = new File(folderPath);
      File[] files = f.listFiles();
      int filesCount = 0, foldersCount = 0;
      for(int i = 0; i < files.length; i++) {
         if(files[i].isDirectory()) {
            //recursively count files in subdirectory
            if(includeSubFolder) {
               int[] filesAndFoldersCount = count(new Criteria(files[i].getAbsolutePath(), includeSubFolder,
extension));
               filesCount += filesAndFoldersCount[0];
               foldersCount += filesAndFoldersCount[1];
            } else {
               foldersCount++;
            }
         } else if(files[i].getName().endsWith(extension)) {
            //file with right extension
            filesCount++;
         }
      }
      return new int[]{filesCount, foldersCount};
   } catch (Exception e) {
      System.err.println("Not a valid directory!");
   }
}

public static void main(String[] args) {
   Scanner in = new Scanner(new BufferedReader(new InputStreamReader(System.in)));
   //the input format is "<fileDirectory>" "<extension>" true/false
   Criteria criteria = new Criteria();
   criteria.folderPath = in.next();
   criteria.extension = in.next();
   criteria.includeSubFolder = in.nextBoolean();
```

```
    int[] filesAndFoldersCount = count(criteria);
    int filesCount = filesAndFoldersCount[0], foldersCount = filesAndFoldersCount[1];
    System.out.println("There are " + filesCount + " files and " + foldersCount + " folders inside folder " +
criteria.folderPath +  "with extension " + criteria.extension);
}
```