**Assignment 2**

1. Why we need packages in java?

Organize code and classes and prevent collisions while coding.

1. What is the default imported package?

Java.lang package is the default.

1. What is Class? What is Object?

Class is the blueprint from which all objects are made of. Object is an instance of that class.

1. Why we need constructor?

To initialize attributes with default values or parameter values.

1. What is the default value of local variable? What is the default value of instance variable?
2. What is garbage collection?

Garbage collection is the process of deallocating useless memory that the program no longer needs. This frees space to create more objects

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

True, Accessible to the same package and subclasses.

1. What is immutable class?

An immutable class has no methods for change the state of the function. Therefor, they only have getters. By nature they are thread safe.

1. What’s the difference between “==” and equals method?
2. What is wrapper class?

Wrapper class is a simple class that has maybe a primitive as it’s only attribute. This in terms wraps the primitive in a class.

1. What is autoboxing?

Autoboxing is when java converts a primitive to a wrapper class.

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

False.

1. Constructor can be inherited, true or false?

False.

1. How to call a super class’s constructor?

Super() or super(arg1,…argn);

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

Object class is the super class of all classes.

1. 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** antra\_assignment2;
* **import** java.io.File;
* **import** java.util.Scanner;
* **public** **class** Main {
* **static** **int** *files*;
* **static** **int** *folders*;
* **public** **static** **void** main(String[] args) {
* // variables
* Criteria crit = **new** Criteria();
* Scanner scan = **new** Scanner(System.***in***);
* //set criteria information
* System.***out***.println("Enter folder name or complete folder path");
* crit.setFolder(scan.nextLine());
* System.***out***.println("Enter t or f for include subfolder");
* crit.setIncludeSubFolder((scan.nextLine().charAt(0) == 't')? **true**:**false**);
* System.***out***.println("Enter File Extention to count. Examples \* for all, .\*, .java, or .txt. One Extention only");
* crit.setExtention(scan.nextLine());
* //time and execute count
* **long** begin = System.*currentTimeMillis*();
* *count*(crit);
* **long** end = System.*currentTimeMillis*();
* **long** time = end-begin;
* System.***out***.println("Time Milli Seconds : " +time);
* }
* **public** **static** **void** count(Criteria crit)
* {
* //couts the total number of files/folders that meet the criteria
* //if file/folder exists then continue
* File file = **new** File(crit.getFolder());
* **if** (file.exists())
* {
* *recursiveCount*(crit, file);
* System.***out***.println("There are "+*files*+" file(s) and "+*folders*+" folder(s) inside folder "+crit.getFolder()+" with extension "+crit.getExtention());
* }
* **else**
* System.***out***.println("File/Folder does not exist." );
* }

* **public** **static** **void** recursiveCount(Criteria crit, File file) {
* //does the counting recursively
* **if** (!file.exists())
* **return**;
* // check if directory exists
* **if** (!file.isDirectory()) {
* //count file
* **if** (*isSuffix*(file.getName(), crit.getExtention())) {
* *files* +=1;
* }
* **return**;
* }
* **else**{
* //count folder
* **if** (file.getName().contains(crit.getExtention())) {
* *folders* +=1;
* }
* //Enables recursion
* **if** (crit.isIncludeSubFolder())
* **try** {
* **for** (File f : file.listFiles()) {
* *recursiveCount*(crit,f);
* }}
* **catch**(Exception e) {
* }
* }
* }
* **public** **static** **boolean** isSuffix(String s, String suffix) {
* **if** (s.length() < suffix.length())
* **return** **false**;
* //System.out.println(s.substring(s.length() - suffix.length()) + " "+ suffix);
* **if** (s.substring(s.length() - suffix.length()).equals(suffix))
* **return** **true**;
* **return** **false**;
* }
* }
* **class** Criteria{
* **private** String folder ;
* **private** **boolean** includeSubFolder;
* **private** String extention;
* **public** Criteria(String folder, **boolean** includeSubFolder, String extention) {
* **super**();
* **this**.folder = folder;
* **this**.includeSubFolder = includeSubFolder;
* **this**.extention = extention;
* }
* **public** Criteria() {
* // **TODO** Auto-generated constructor stub
* folder = "/";
* includeSubFolder = **true**;
* extention = ".java";
* }
* **public** String getFolder() {
* **return** folder;
* }
* **public** **void** setFolder(String folder) {
* **this**.folder = folder;
* }
* **public** **boolean** isIncludeSubFolder() {
* **return** includeSubFolder;
* }
* **public** **void** setIncludeSubFolder(**boolean** includeSubFolder) {
* **this**.includeSubFolder = includeSubFolder;
* }
* **public** String getExtention() {
* **return** extention;
* }
* **public** **void** setExtention(String extention) {
* **this**.extention = extention;
* }
* }