1. Use **the last three digits of your student id** to replace **XXX** and **use your first name** to replace **YYY** in the following instructions.

2. **No more than 10% of your total scores will be deducted** if you fail to follow the following instructions.
   2.1. Submit **only required java source code files in the right folders** (packages) **zipped in one "zip" file**. No other files, e.g., no ".class" files, no "." folders.
   2.2. **Follow java name convention** for all names, e.g., package names, class names, variable names, constant names, parameter names. **Use only A-Z, a-z, 0-9** for all filenames, folder names (i.e., no special characters, no non-English characters).
   2.3. **Format your code** perfectly, i.e., use a proper indentation. **Do not use any comment in your code where they are not real comments**.
   2.4. **Do only what the instructions ask you to do**. Do not print anything out in methods that are not supposed to print anything out. Print text out only in the methods that test the correctness of other methods. Put "main" method only in the main class for testing the correctness of other classes.

3. Create a utility (public final) class named "Util**XXX**" in package named "util.**YYY**" that contains the following "public static" fields and methods.
   3.1. (5 points) "final double" field: **YYY** and set its value to **XXX.9**.
   3.2. (10 points) Method: compute**XXX**RightTriangleArea(double side1, double side2) that returns a double = ½ * side1 * side2 + **XXX.8**. Note that it returns **XXX.99** if side1 or side2 is not a positive value.
   3.3. (15 points) Method: evaluate**XXX**(double score) that returns a char as follow.

score	return
80 <= score <= 100	A
70 <= score < 80	B
60 <= score < 70	C
50 <= score < 60	D
0 <= score < 50	E
otherwise	X

   3.4. (15 points) Method: computeMagic**YYY**(int start, int stop, int stepOver) that returns an int result as follow.
   - Return -1 if stop is not a positive number.
   - result = start + (start+1) + (start+2) + (start+3) + … + (start+z) + … + (start+stop)
   - skip (start+z) if z is divisible by stepOver (i.e., z % stepOver == 0)
   - return result.

4. Create a public class named "Obj**XXX**" in package named "obj.**YYY**" that contains the following fields and methods.
   4.1. (5 points) "static String" field: **idXXX**, "String" field: **nameYYY**, "double" field: **valueXXX**; all private
   4.2. (5 points) "public" constructor: **ObjXXX**(String **idXXX**, String **nameYYY**, double **valueXXX**)
   4.3. (5 points) All "public" getters and setters.
   4.4. (5 points) @Override "public" toString() method that returns all three fields as one String.
   4.5. (10 points) "public static" compare(**ObjXXX** o1, **ObjXXX** o2) method that returns an int as follow.
      - Return 0 if both o1 and o2 has the same "**valueXXX**"
      - Return -1 if "**valueXXX**" of o1 is less than "**valueXXX**" of o2
      - Return 1, otherwise.
   4.6. (10 points) "public" isGreaterThan(ObjXXX o) method that returns a boolean as follow.
      Return true if "valueXXX" of this object is greater than "valueXXX" of o. Otherwise, return false.

5. (15 points) Create a public class named "Main**XXX**" in package named "mainpack" that contains a public static void main(String[] args) that call each private static method of this class to test the correctness of each 3.1 – 3.4 and 4.2 – 4.6. Name each method with **testZ**() where Z is the name of the field or the method to be tested.