THOMPSON RIVERS UNIVERSITY

COMP 1130 Computer Programming I Final Review Questions 3

- **Assume the following are always available:**
    - all packages, libraries, and classes are already **import**ed
    - **display () / displayln ()** – in place of System.out.print() / System.out.println ()
    - **inputStr**(String prompt) – which displays a prompt on the console, and returns the user's complete keyboard input as a String (using .nextLine(), not .next())
    - String-to-number conversions using:
      - double x = **Double.parseDouble("3.14");** converts "3.14" to 3.14
      - int x = **Integer.parseInt("42");** converts "42" to 42

Short Programming Questions

[1 mark (0.5 each)]

1. Write both the **display()** and **displayln()** methods. Both methods have a single String parameter (which is displayed to the console) and both return nothing.
   Consider the following sample calls,

```
display ("show this");
displayln ("& \"this\" ");

output is:
      show this & "this"
      _
```

[1 mark (0.5 each)]

2. Calling only the methods above, write two (2) more methods that overload **displayln(),** one that has a single **int** parameter, and one that has no parameters and only outputs a newline to the console. Both also return nothing.
   Consider the following sample calls,

```
displayln (42);
displayln ();

output is:
      42
      _
      _
```

[1 mark]

3. Write the method **inputStr()**, it has a single String parameter, which is displayed to the console, and returns a String. The return String contains the user's <u>full</u> input.
   The method must perform <u>all</u> necessary operations with a Scanner class object).
   Consider the following sample call:

```
String phr = "";

phr = inputStr("What? ");

displayln (phr);
```

[1 mark]

4. Rewrite the method **init()**, such that it matches the form of the second call.
   The new form has a single parameter, the size of the array, and returns an array of int.

```
public static void main (String[] args)
 {
      double[] listA = new double[200];
      double[] listB;

      init(listA);
      listB = init(200);
 }// end of main()

 public static void init (double[] ray)
 {
      for (int i=0; i<ray.length; i++)
      {
           ray[i] = 0.0;
      }
}// end of initArray()
```

[1 mark]

5. Write the method **inputInt()**, it has a single String parameter, which is displayed to the console, and returns an int. *You are <u>not permitted</u> to use the Scanner class, System.out.print(), nor System.out.println().*
   Consider the following sample call:

```
int val = 0;

val = inputInt("Value? ");

displayln (val);
```

[2 marks]

6. Write the method **multiConcat()** that has two parameters: a String and an integer, and returns a String which contains the String repeated that many times. If the String parameter is empty ("", or length = 0) OR the integer value <= 0, then a blank String ("") is returned.
   Consider the following sample calls, which display, respectively: **qqqqq** and **BarkBarkBark**

```
String line = multiConcat("q",5);
displayln (line);   // display qqqqq
line = multiConcat("Bark",3);
displayln (line);   // display BarkBarkBark
line = multiConcat("House",-5);
displayln (line);   // display empty string
```

Errors in Code – *Fix it!* – Questions

[1 mark]

7. You were given this code segment, that tests the two techniques for producing integer random numbers. But the compiler is indicating a "possible lossy conversion" error—*fix it!*

```
 // two techniques to generate
 // integer random numbers, from 1..12
Random rng = new Random();
int rndA =  rng.nextInt(12) + 1;
int rndB =  (Math.random()*12)+1;
displayln (rndA+" "+rndB);
```

[2 marks]

8. The following main() method is written to test the method **difference()**.
   It has a compiler error in the method, indicating an initialisation error—*fix it!*
   After the program is fixed, it compiles successfully, but crashes with a run-time "out of bounds" error—*fix it!*

```
public static void main (String[] args)
{
       int[] rA = { 22, 10, 5, -38, 72, 82 };
       int[] rB = { 8, 15, -33, 66, 13 };

       int[] subtract = difference (rA, rB);
}// end of main()

public static int[] difference (int[] a, int[] b)
{
       int[] diff;

       for (int i=0; i<a.length; i++)
       {
              diff[i] = Math.abs(b[i] - a[i]);
       }

       return (diff);
}// end of difference()
```

[2 marks]

9. The method **power()** is provided below, intended to calculate **x**^{y} , such that: 2^3 = power(2,3) = 2x2x2 = 8
   It compiles, but never calculates correctly (always returns zero?)—*fix it!*
   Also, rewrite the method, replacing the inner **while** loop as a **for** loop.

```
public static int power (int x, int y)
{
       int res = 0;

       int count = 0;
       while ( count < y )
       {
            res *= x;
            count++;
       }

       return (res);
}// end of power()
```

[2 marks]

10. The switch statement below has a compiler error with **switch(direction)**, why?
    Also, when it runs, a logic error (everything seems to be 'W' ?!)--*fix it!*

```
double direction = 2.0;
char compass;

switch (direction)
{
    case 1:
         compass = 'N';
    case 2:
         compass = 'E';
    case 3:
         compass = 'S';
    case 4:
         compass = 'W';
         break;
    default:
         compass = '?';   // unknown dir.
         break;
}
```

Long Programming Questions

[2 marks]

11. Using the basketball point scores data, write the code segment that creates a bar chart comparing the scores for five (5) games between two teams, with "+" for team1 and "-" for team 2.
    (note: there is no input from the user, and the output is only the bar graphs)

    Assume the following arrays are declared at the top of the main(),

    ```
       int[] team1 = { 24, 54, 65, 28, 90 };   // games 1..5 for both teams
       int[] team2 = { 35, 41, 61, 70, 22 };
    ```

    The bar graph is displayed as,

    ```
    Game 1:
    24: ++++++++++++++++++++++++
    35: -----------------------------------
    Game 2:
    54: ++++++++++++++++++++++++++++++++++++++++++++++++++++++
    41: -----------------------------------------
    Game 3:
    ...
    and so on
    ```

[3 marks]

12. Assume the String class method **.equals()** and **.compareTo()** do not exist. You still need it, so decide to write your own equivalent **equals()** method, that has two String as parameters, and returns a boolean *true* (if the Strings are the same) or *false* (if the Strings are not the same).
    The basic rules,
    - if the two strings have different lengths, they are not equal → false
    - if both strings are empty (length zero), they are equal → true
    - if all characters at the same position in the two strings are the same, strings are equal → true
      (which also means if any character is different in the same position, strings are not equal →false)

    Consider the following sample calls,

```
String strA = "word",
       strB = "words",
       strC = "works";
boolean res;

res = equals(strA,"word"); // true
res = equals("","");       // true
res = equals(strA,strB);   // false
res = equals(strB,strC);   // false
```

[4 marks]

13. Write the program that performs a guessing game.
    Use arrays, if_else, switch, for, while, inputStr(), charAt(), *casting*, and anything else you consider necessary.

    The computer 'thinks' of a symbol (!, @, #, $, %, or &), and asks the user to guess which one,
    - it randomly picks one of the characters
    - prompts the user to "Guess which symbol: !, @, #, $, %, or &"
    - if the user's input matches the computer, it displays "You Guessed It!"; otherwise, it displays "Sorry. It was X" (where X is the symbol the computer picked)

[5 marks]

14. Write the program, that acts as a simple "look up" database, to find a person's phone number based on their name. *Use all the methods and concepts expressed in the previous questions, and your knowledge of Java.*
    Assume the following arrays are already populated as globals in the program.

```
String[] people = { "Bob Smith", "Alice Thom", "Ashnav Singh", "Lin Kim",  "Guy Perez" };
String[] number = { "828-1121",  "579-9950",   "852-1953",     "555-1111", "372-0010"  };
```

    Rules,
    - the person's name for look up can be entered in upper, lower, or any case, and it is still found
    - if the person's name can not be found, the program displays "unknown"
    - the program continues until the name "nobody" is entered

    Consider the following user interaction (user input is in **bold**),

```
Name? Bob SMITH
Phone: 828-1121
Name? Betty Pritty
Phone: unknown
Name? ASHNAV SINGH
Phone: 852-1953
Name? Nobody

program terminated
```

[8 marks]

15. Tabulate the outcomes of two (2) rival basketball teams playing against each other.
    The two teams played 5 games each other, and the points each team scored per game are stored in two (2) arrays: team1[], team2[], and the team names are stored in teamNames[]

    Using these arrays, write the main() that displays,
    - table showing the point scores, and which team won each game (highest score wins)
    - highest (max) points scored, per team
    - number of games won, per team
    - average points scored per game, per team (total points / number of games), to 1 decimal place

    Again, as seen in a previous question, assume the following are declared at the top of the main(),

```
int[]    team1     = { 24, 54, 65, 28, 90 };  // games 1..5 for both teams
int[]    team2     = { 35, 41, 61, 70, 22 };
String[] teamNames = {"TRU Ballers","Java Coders"};  // names of teams 1 & 2
```

The tabulation and results output,

```
Game 1: TRU Ballers @ 24, Java Coders @ 35, winner Java Coders
Game 2: TRU Ballers @ 54, Java Coders @ 41, winner TRU Ballers
Game 3: TRU Ballers @ 65, Java Coders @ 61, winner TRU Ballers
Game 4: TRU Ballers @ 28, Java Coders @ 70, winner Java Coders
Game 5: TRU Ballers @ 90, Java Coders @ 22, winner TRU Ballers

TRU Ballers max points: 90
Java Coders max points: 70

TRU Ballers wins: 3 games
Java Coders wins: 2 games

TRU Ballers average score: 52.2
Java Coders average score: 45.8
```