School of Computer Science and Information Technology. RMIT

COSC2531 Programming Fundamentals
Assignment 1
Due date: 9:00AM 29th August, 2016
20% of course assessment

Q1. (7 marks)

Assume two integer variables, x and y, that contain data.
Show *by using a truth table* which, if any, of the following IF statement tests is equivalent to:
if (!(x == 30 && y == 10))

- if (x != 30 || y != 10)
- if (x == 30 && (!(y == 10)))

Q2. (13 marks)

Write a Java class called *Game* that plays the game 'paper, scissors, rock' with a user.
A game consists of a user-defined number of rounds.
A single round of the game works as follows:

> The user and the program each choose a number: 0 (for 'paper'), 1 (for 'scissors') or 2 (for 'rock'). The program uses a random number generator to make its choice (which it then prints out to assist in marking the program).
> The result of the round is as follows:

Player choice combination	*Game result*
Same item selected	A draw
Paper + Scissors	Scissors wins: scissors cut paper
Paper + Rock	Paper wins: paper wraps rock
Rock and Scissors	Rock wins: rock blunts scissors

So for example, if the user chose paper and the program chose scissors then the program wins that round. But if the program chose paper and the user chose scissors, then the user wins that round.

Example output of a single round of the game is:

```
Enter 0, 1 or 2 for paper, scissors, rock
Computer Choice: 1

My choice was :  2
Rock blunts scissors: I win
```

Your *Game* class should contain:

- a public integer instance variable called *numRounds*, that records how many rounds of the game have been played.
- a public integer instance variable called *roundsWon*, that records how many

rounds of the game have been won by the human player.
- a public void method called *playGame()*, which plays the game

Method *playGame()* should ask the user how many rounds to play, and then play that many rounds of the game. Variables *numRounds* and *roundsWon* should be correctly updated by the method.

Apart from the main method, rest of the method should be non-static method.

Coding style will be part of the marking criteria.

Submission:

1. Due date: 9:00AM 29th August, 2016. Representing 20% of course assessment

2. You will need to submit two electronic files in **one zip** (not rar, not 7z or other compression format) file through weblearn system in blackboard.
   - a PDF file containing the answers to questions 1 and screenshots of a successful execution of the question 2 named *ass1.pdf* (no Word files)
   - a Java file called *Game.java* containing the solution to question 2.

3. Please note the Java file must be a Java source files (e.g. Game.**java**) and can run under Eclipse and command prompt, not a class file (e.g. not Game.**class**). We can't mark a class file. Please put the screenshots of the successful run in the ass1.pdf file together with question 1.

4. **One** student only allows **one** submission. Errors caused by multiple submissions will be your own responsibility in finalizing scores.

Standard warning:
This assignment must be your own work. Any submission found in whole or in part to be plagiarized will be given zero marks and will not count towards your assessment. If an award of zero marks means that you do not pass the assignment hurdle then you will fail the course.

Marking Guide (Total 20 marks)

Question 1: 7 marks

Details should be displayed in a step by step fashion of the truth table to get full marks.

Question 2: 13 marks

- 4 marks for functional working code
- 4 marks for capturing error input and allow user to correct them to differentiate the efforts of some students who spend extra efforts to make code perfect.
- 5 marks for Coding Requirement
  -- meet all the requirements of specification requirement like no magic variables, indentations etc. Submit in right zip format
  -- conduct java doc style comments. .

  For all assignments, remember they are a learning process rather than punishing you for the mistakes. We will be soft and very considerable in pass and fail issues.