## University of Cape Town Department of Computer Science

## Computer Science CSC115F

## Class Test 2

- Answer all questions.
- All questions that refer to elements of programming make reference to the Java programming language as studied in class.
- Good luck !

| Marks: 40 | -Approximate marks per question are <br> shown in brackets <br> Time: 40 minutes |
| :--- | :--- |
| - The use of calculators is permitted |  |

This paper consists of 1 questions and 3 pages (including this cover page).

| Mark Allocation |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Quest | Marks | Internal | External | Quest Marks | Internal | External |
| 1 [10] |  |  |  |  |  |  |
| Total |  |  |  | Total |  |  |
| Grand Total |  |  |  |  |  |  |
| Final Mark |  |  |  |  |  |  |
| Internal Examiner: |  |  |  | External Examiner: |  |  |

## Section 1. Selection

## Question 1. [10 marks]

a) Write a method for the NAND operation, which is false if and only if both arguments are true, using only "if" statements, tests for equality where needed and "return" statements. Use the following template:

```
public boolean nand ( boolean a, boolean b )
{
```

\}
b) The following code fragment sets prerequisites for a hypothetical set of courses. Rewrite the code to use a "switch" statement, with the order of cases optimised to use a minimal number of "break"s.

```
if (course == 201)
{
    needMATH101 = true;
}
else if (course == 202)
{
    needPHYS201 = true;
    needMATH101 = true;
}
```

$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
c) The intent of the following erroneous code fragment is to flag an error if a mark does not lie in the range $0-100$.

```
if (myMark>=0)
    if (myMark>100)
        invalidMark = true;
else
    invalidMark = true;
```

Without changing the intent, fix and rewrite the code in 3 different ways as indicated below:-
i. The order of statements remains the same - insert a single pair of curly braces to prevent the dangling "else" problem.
ii. Swap the "if" and "else" portions of the outer "if" statement to prevent the dangling "else" problem - the result should be an "if ladder".
iii. Combine the conditions with a boolean operator, thus avoiding duplication of the assignment statement and ultimately avoiding the dangling "else".
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$

