UNIVERSITY OF JOHANNESBURG

FACULTY OF MANAGEMENT
JUNE EXAMINATION

DEPARTMENT OF APPLIED INFORMATION SYSTEMS

MODULE : **DEVELOPMENT SOFTWARE I**

CODE : **DSW01B1**

DATE : 18th NOVEMBER 2014 EXAMINATION

DURATION : 3 HOURS

TIME : 08h30

TOTAL MARKS : 100

EXAMINER : MRS J CHETTY

(EXTERNAL) MODERATOR(S): MRS N PAYNE

NUMBER OF PAGES : 6 PAGES

INSTRUCTIONS TO CANDIDATES:

- Question papers must be handed in.
- Section A is a closed book assessment. Section B is an open book assessment.
- Read the questions carefully and answer only what is asked.
- Number your answers clearly.
- Write neatly and legibly
- Structure your answers by using appropriate headings and sub-headings.
- Calculators may be used
- Answer the multiple choice questions on the supplied answer sheets
- The general University of Johannesburg policies, procedures and rules pertaining to written assessments apply to this assessment

This paper consists of **Section A** and **Section B**. Section A consists of short questions related to the theory of C#. This section must be completed in the examination answer sheets provided to you. Section B is a problem that must be solved using C# using the Visual Studio environment.

SECTION A [40]

QUESTION 1 (10)

Provide an answer of true or false for the following questions.

1.1 Operator precedence rules ensure that addition and subtraction always take place prior to multiplication, division, or remainder in an expression.

1.2 The equal sign (=) is the C# assignment operator.

1.3. someValue++ is an example of the use of the prefix increment operator.

1.4 When if-else statements are nested, each else is always paired with the most recent unpaired if.

1.5 When loop structures are nested, the inner loop must be entirely contained within the outer loop; loops can never overlap.

1.6 In a switch statement, the keyword return usually terminates each case.

1.7 In C#, arrays are objects.

1.8 An escape sequence is used to represent special characters such as a carriage return or a tab, and always begins with a backslash.

1.9 A declaration for a method that receives two or more arguments must list the type for each parameter separately only if the parameters have different types.

1.10 The type of an argument in a method call must exactly match the type of the corresponding parameter specified in the method declaration.

QUESTION 2 (10)

Making use of the multiple choice answer sheet provided to you answer the following multiple choice questions.

2.1 A ____ programming language allows you to use a vocabulary of reasonable terms such as “read,” “write,” or “add” instead of the sequence of on/off switches that perform these tasks.

a. high-level
c. low-level
b. machine-level
d. switch-level

2.2 Programmers use a computer program called a(n) ____ to translate their high-level language statements into machine code.

a. prompt
c. compiler
b. IDE
d. JIT

2.3 When you write a(n) ____ program, you use your knowledge of a programming language to create and name computer memory locations that can hold values, and you write a series of steps or operations to manipulate those values.

a. object-oriented
c. event-driven
b. component-oriented
d. procedural

2.4 For convenience, the individual operations used in a computer program often are grouped into logical units called ____.

a. attributes
c. classes
b. methods
d. structures

2.5 A(n) ____ is a construct that acts like a container to provide a way to group similar classes.

a. namespace c. object
b. method d. black box

2.6 If a value is given to a variable at the time it is declared, this is called a(n) ____.

a. definition c. assignment
b. allocation d. initialization

2.7 A while loop is a(n) ____ loop.

a. posttest c. intest
b. pretest d. fortest

2.8 In C#, an array subscript must be a(n) ____.

a. float c. integer
b. char d. double

2.9 To use the Sort() method, you pass the array name to ____.

a. List.Sort() c. Array.Sort()
b. Collections.Sort() d. Sort.Sort()

2.10 Methods can ____, or call, other methods; that is, a program uses a method's name and the method performs a job for the class.

a. instantiate c. inherit
b. overload d. invoke

QUESTION 3 **(10)**

Making use of the answer sheet provided to you fill in the missing word (s) given the following questions.

3.1 A computer _____ is a set of instructions that you write to tell a computer what to do.

3.2 Named computer memory locations are called _____ because they hold values that might vary.

3.3. A data item is _____ when it cannot be changed after a program is compiled.

3.4 Giving a value to a variable, as in the statement someVariable = 5, is called a(n) _____.

3.5 Programmers often use _____, a tool that helps them plan a program's logic by writing plain English statements.

3.6 The _____ structure, an alternative to nested if statements, tests a single variable against a series of exact matches.

3.7 You can use a(n) _____ loop to execute a body of statements continuously as long as some condition continues to be true.

3.8 A method _____ is a block of statements that carry out the method's work.

3.9 A value that a user must supply to stop a loop (for example, 'Y' or 'N') is called a(n) _____ value.

3.10 A(n) _____ array has the same number of elements as an existing array and uses the same subscript to access additional information.

QUESTION 4 (10)

Trace through the following code and determine the output for each.

4.1
```
for (int x = 9 ; x >=0 ; x--)
        {
                y = x + 5;
                Console.WriteLine("The output for y is {0} and x is = {1}", y, x);
        }
```
(3)

4.2
```
static void Main()
        {
                double myPurchase = 15.90;
                DisplaySalesTax(myPurchase);
                DisplaySalesTax(45.65);
        }

private static void DisplaySalesTax(double saleAmt)
        {
                double tax;
                const double RATE = 0.07;
                tax = salesAmt * RATE;
                Console.WriteLine("The tax on {0} is {1}", saleAmt.ToString("C"),
tax.ToString("C"));
        }
```
(4)

4.3
```
int [] nbrs = new int [5] { 12, 5, 22, 16, 18};
for (int a = 0 ; a < 5 ; a++)
        {
                nbrs [a] = nbrs [a] + (nbrs [a] +5);
                Console.WriteLine("The amount now is {0}", nbrs [a]);
        }
```
(3)

SECTION B [60]

Solve the following problem by writing a C# program. The final product must be saved to the uLink folder provided to you.

The Buggy-Me-Along car company has requested you to develop a user-driven application for them. They would like an application that will allow them to perform the following options / choices:

1. Add vehicles to the system
2. Calculate discount
3. Determine the lowest vehicle's / highest vehicle's price
4. Find a vehicle's price
5. List all cars in a category
6. Exit

In order to accomplish this task, the following specifications have been provided to you. Good luck!

(a) You need to include the following arrays in order for the application to work efficiently: regNbr, price, carCat

Array name	**Description**
regNbr	This is an array that holds 10 elements. Each element is the registration number of the vehicle
price	This is an array that holds elements related to the cost of each vehicle found in regNbr array
carCat	This is an array that holds elements related to the category of each vehicle (either A, B or C) found in regNbr array

(b) The user-driven application must consist of a menu with 6 options / choices, seen above. The application menu must continue to display until the user decides to exit. You must make use of the case / switch statement to control which option on the menu is chosen.

(c) Each option numbered 1 to 6 must represent the operations attached to that option. The operations must be structured within methods. In other words, you must call a method from the case / switch statement and the body (operations) of each option must be inside the method. You may need to pass parameters to the method in order for the operation to work properly.

a. Option 1:

the code within this option should call a method called AddVehicle(). This method must ask the user to enter the vehicle registration number, the cost of the vehicle and the category that the vehicle falls under

b. Option 2:

The code within this option should call a method called FindDisc(). This method must determine the discount offered to a customer purchasing the vehicle. The method must determine the discount for every vehicle in the array, based on the following table:

Price range	**Discount percentage off the cost of the vehicle**
Less than R50 000.00	No discount
Less than R100 000.00	10% discount
Less than R150 000.00	15% discount
From R150 000.00 up	20% discount

This option must also print out the vehicle registration number, together with the new cost of the vehicle.

c. Option 3:

The code within this option should call a method called FindLowHigh(). This method must determine which vehicle has the lowest price and which vehicle has the highest price. Again, the vehicle registration number as well as the price must be displayed.

d. Option 4:

The code within this option should first ask the user to enter a vehicle registration number. This number must be passed as a parameter to a method called FindPrice(). This method must search all the vehicles and locate the vehicle registration number that the user entered. It must then display the price of the vehicle.

e. Option 5:

The code within this option should call a method called FindCatA(). This method should locate all the vehicles that are category A. a list of the vehicles in category A should be displayed.

f. Option 6:

This option means that the user wants to exist the application. The code should reflect this.