ދިވެހިރާއްޖޭގެ ޤައުމީ ޔުނިވަރސިޓީ

THE MALDIVES NATIONAL UNIVERSITY

FACULTY OF ENGINEERING TECHNOLOGY

BACHELOR OF CIVIL ENGINEERING

…..

COMPUTING FOR ENGINEERING

SUBJECT OUTLINE

April 2014

SUBJECT OUTLINE

COMPUTING for ENGINEERS

1	Subject Name	Computing for Engineers
2	Subject Code	**……**
3	Learning hours Credit Points Contact hours	120 12 14 x 5 hours lecture/laboratory
4	Minimum Qualification and Experience Required by Instructors/Lecturers	Civil Engineering Degree with 2 Years Relevant Experience
5	Maximum Class Size	Lectures: Indeterminate Tutorials: 20 students
6	Pre-requisites	None
7	Co-requisites	None
8	Expected learning outcomes	Upon completion of this course, with respect to low-rise residential construction, learners will be able to: • Design, program, document and test Visual Basic for Applications (VBA) programs that accurately comply with a given specification. • Use applications such as spreadsheets and databases to effectively solve computational problems in science, engineering, commerce and humanities. • Gain an understanding of the limitations/advantages of some of the emerging IT technologies, and how to exploit them in solving engineering problems.
9	The Mode(s) of Delivery (i.e. Lectures, Seminars, Workshops, Distance Education, Group Work etc.)	This subject will consist of series of lectures followed by laboratory classes.

10a.	Curricular Content in Terms of Topics and a Summary of Content Planned for each Lecture/Tutorial Session/Seminar etc.

This subject is an introductory first year computing course taught by the Faculty of Engineering Technology. By the time you finish the course you should understand the concepts and techniques covered by the syllabus and have developed skills in applying those concepts and techniques to the solution of appropriate problems. The subject aims are;

- Introduction to popular applications (such as spreadsheets and databases) to solve computational problems
- Introduction to procedural programming (using Visual Basic) to solve engineering tasks
- Introduction to some of the current and emerging Information Technologies, with the aim to understanding their limitations/advantages and how they can be exploited to innovatively and effectively solve problems.

Introduction to Visual Basic .NET (extension material, designed to challenge "good" students but not required for a bare pass).

10b.	Course Outline
Topic	Lecture
Spreadsheets	1, 2 & 3
Problem Solving and Programming	4, 5, 6, 7 & 8
Numerical Computing	9, 10, 11 & 12
Technology: Past, Current & Future	13 & 14

10c.	Course Syllabus

LECTURE 1, 2 & 3

SPREADSHEETS

Basics of Spreadsheets

Why Use a Spreadsheet?
Problem Solving
Key Concepts
Number Representation and Limits
Charts
Organising Data: sorting and filtering
Useful functions and Features

Data Analysis using Spreadsheets 1

Data Analysis
Pivot Tables
Simple Statistics
Histogram
Correlation
Fitting Equations to Data
Presenting Charts
Solving Single-Variable Equations
Goal Seek examples: black holes and ballistics

Data Analysis using Spreadsheets 2

Finding Optimum Solutions
Matrix Operations
Solving Equations
Financial Functions, Look-up Functions, Testing and Validating Spreadsheets

LECTURE 4, 5, 6, 7 & 8

PROBLEM SOLVING & PROGRAMMING

Introduction

The Language of OpenOffice.org Basic

Terminology – Plate, ceiling joist, trimmer, hangers, hanging beams, ceiling dog, framing anchor, blocking and insulation.

Program Lines , Markers (Identifiers).

Implicit Variable Declaration , Explicit Variable Declaration From a Set of ASCII Characters to Unicode, String Variables, Specification of Explicit Strings, Integer Variables Long Integer Variables , Single Variables , Double Variables Currency Variables , Floats , Specification of Explicit Numbers

Defining Arrays Defining values for arrays Accessing Arrays , Array Creation, value assignment and access example

Local Variable, Public Domain Variables, Global Variables, Private Variables

Defining Constants Scope of Constants Predefined Constants

Mathematical Operators, Logical Operators, Comparison Operators

If...Then...Else, Select...Case, For...Next, For Each, Do While...Wend

Programming Example: Sorting With Embedded Loops

Subprograms, Functions, Terminating subprograms and Functions Prematurely Passing Parameters, Optional Parameters

Recursion, The On Error Instruction, The Resume Command, Queries Regarding Error Information, Tips for Structured Error Handling

Type...End Type, With...End With

Runtime Library

Implicit and Explicit Type Conversions, Checking the Content of Variables

Working with Sets of Characters Accessing Parts of a String Search and Replace Formatting Strings

Specification of Date and Time Details within the Program Code Extracting Date and Time Details, Retrieving System Date and Time

Administering Files Writing and Reading Text Files Displaying Messages, Input Box For Querying Simple Strings, Beep, Shell, Wait and Wait Until , Environ

Introduction to the API

Properties, Methods , The supports Service Method Debug Properties, API Reference, Creating Context-Dependent Objects, Named Access to Subordinate Objects, Index-Based Access to Subordinate Objects, Iterative Access to Subordinate Objects

Working with Documents

This Component Basic Information about Documents in OpenOffice.org, Creating, Opening and Importing Documents, Document Objects, Details about various formatting options

Text Documents

Paragraphs and Paragraph Portions, The Text Cursor, Searching for Text Portions, Replacing Text Portions, Tables Text Frames, Text Fields, Bookmarks

Spreadsheet Documents

Spreadsheets, Rows and Columns, Cells and Ranges, Formatting Spreadsheet Documents, Cell Ranges, Searching and Replacing Cell Contents

Drawings and Presentations

Pages, Elementary Properties of Drawing Objects, An Overview of Various Drawing Objects, Grouping Objects, Rotating and Shearing Drawing Objects, Searching and Replacing, Working With Presentations

Charts (Diagrams)

Details about various formatting options, Title, Subtitle and Legend, Background, Diagram, Wall and Floor Axes, Properties of Axes Grids, Axes Title, 3D Charts Stacked Charts, Line Charts Area Charts, Bar Charts Pie Charts.

Databases

Queries, Iteration of Tables, Type-Specific Methods for Retrieving Values, The Result Set Variants, Methods for Navigation in Result Sets, Modifying Data Records

Dialogs

Creating Dialogs, Closing Dialogs, Access to Individual Control Elements, Working With the Model of Dialogs and Control Elements, Name and Title Position and Size Focus and Tabulator Sequence, Multi-Page Dialogs, Parameters, Mouse Events Keyboard Events, Focus Events, Control Element-Specific Events Buttons, Option Buttons, Checkboxes Text Fields, List Boxes

Forms

Determining Object Forms. The Three Aspects of a Control Element Form , Accessing the, Model of Control Element Forms, Accessing the View of Control Element Forms, Accessing the Shape Object of Control Element Forms, Buttons, Option Button, Checkboxes Text Fields, List Boxes, Tables

LECTURE 9, 10, 11 & 12

NUMERICAL COMPUTING

Modeling and visualization using MATLAB.

LECTURE 13 & 14

TECHNOLOGY: PAST, CURRENT & FUTURE

Computer Architecture

Goal Seek examples: black holes and ballistics
Sequential Computing
Logical View of the Architecture
The Computer as an Information Processor
Parallel & Distributed Computing
Embedded Systems
RFID Tags

Digital Revolution and Network Computing

Why Everything's Gone Digital, Bits and Bytes, Encoding
Numbers, Characters, Instructions; Digitising Text
Encoding Sounds and Pictures, Preservation
Encoding/Decoding
What is the Internet, Anyway?, Origins and Development
Communication Types
How does the Internet Work?
How does a LAN work?
Connecting a computer to the Internet
Web Services
Cloud Computing

Computer Security

Cloud Computing
The need for security, firewalls

- Cryptography, symmetric key systems
- Public key systems
- Digital Signatures
- Error Detection and Correction
  - Parity check
  - EAN product codes
  - Hamming codes

Artificial Intelligence

- Philosophical Start
- What Is Artificial Intelligence?
- Knowledge Representation & Reasoning
- Rule-based Reasoning
- Expert Systems/ Knowledge Based System
- Machine Learning
- Decision Trees
- Seeing, Hearing, and Understanding
- AI in Action
- Can Computers be Creative?
- Summary

11. Assessment Methods and Grading Criteria.

Assessment will involve the following:

Task	Value
Assignment 1	10%
Assignment 2	10%
Assignment 3	10%
Lab Exercises	10%
Mid-Session Test	10%
Final Written Exam	50%
Total	100%

To obtain a passing grade in the subject students must obtain an aggregate mark of 50% or better and attend all site visits.

Laboratory

Laboratory Classes give you a chance to **practise** problem solving and programming skills on small, well defined examples. The examples have been chosen to highlight particular aspects of problem solution, and will give you enough grounding in problem solving to assist you in completing your assignment work. Your tutor will be present in your Lab Class to answer any questions you may have.

All lab class exercises must be shown to your tutor during your scheduled lab class. The lab exercises will be available during the session via the course's web page. **Students are advised not to fall behind in laboratory work**. In order to get a mark for a lab exercise you must show your tutor a completed or substantially completed solution before the end of your scheduled lab class in the week that the exercise is released. To gain a mark you must at the very least: show your tutor a partial solution at the end of your scheduled lab class in the week that the exercise is released. If you fail to do so, you will not receive a mark for that particular lab exercise. If you cannot complete the exercise by the end of the scheduled lab, you may complete it in your own time and show it to your tutor in the next scheduled lab class. Your tutor will only mark it at the very beginning of the next scheduled lab class. Note that, **after this time, you will not receive a mark for that particular lab exercise**.

Please note that at the time of marking your lab exercises, your tutor will ask you to solve other similar problems. You need to demonstrate that you are able to solve lab exercises and related problems, in order to receive any marks for your lab work. In other words, copying lab solutions is pointless!

If you are unable to attend your scheduled lab class due to illness or misadventure, then you should apply for special consideration and an extension may be granted. You can only have your lab work **marked** during **your** scheduled **lab class**. It is **not** possible to have your work marked in another class.

Your weekly lab class is 2 hours long and it starts in Week 2 of session.

Assignments

Assignments give you the chance to practise what you have learned on relatively large problems (compared to the small exercises in the Lab Classes). Assignments are a very important part of this course; therefore it is essential that you attempt them yourself. There will be three assignments:

- Assignment 1 will be due in Week 5 (requirements will be available in week 2 or 3).
- Assignment 2 will be due in Week 10 (requirements will be available in week 8).
- Assignment 3 will be due in Week 14 (requirements will be available in week 11).

Assignments are to be completed in your own time. To maximize the learning benefits from doing assignments, it is essential that you start work on assignments early. Do not leave your assignments until the last minute. If you submit an assignment late, the maximum available mark is reduced by 10% per day that it is late.

Assignment 1: the aim in this assignment is to carry out "smart" online research on a given topic. You need to find the required information available online on a given topic, properly organize this information, and produce an online report in the form of a web page(s). There are no explicit marks for this assignment however the concepts will form an integral part of the Mid-Session Test and you must complete it.

Assignments 2 and 3: the aim for these assignments is to develop skills in all stages of the programming process: start by understanding the specification; design a method to solve the problem; refine this design; implement and document the design in a source file that consists of both the documentation and the program code; and test extensively to validate that the program meets the specification.

Mid-semester Examination

A Mid-Session Test will be held in Week 7 during your scheduled lab class.

Final Examination

The final written examination will be held during the examination period. It will examine all material covered in the course, but will emphasize the material in the second half of the course.

12.	List of Reference Materials. **TEXTBOOK** 1. Gottfried, B (2002). *Spreadsheet Tools for Engineers Using Excel.* McGraw Hill. 2. Shepherd, R (2004). *Excel VBA Macro Programming.* McGraw Hill