**DCIT 101**

**INTRODUCTION TO COMPUTER SCIENCE**

**STUDY GUIDE**

**For Undergraduates Level 100**

2021/22 Academic year

**Acknowledgements**

Many thanks to the Teaching Assistants in the who played a critical role in the development and editing of this study guide and the development of the course slides and presentation of the course module on the University of Ghana Sakai Learning Management System

**CONTENTS**

[COURSE DESCRIPTION 6](#_Toc442802273)

[1. COURSE WEBSITE 6](#_Toc442802274)

[2. INSTRUCTORS 6](#_Toc442802275)

[3. Support Contact Information 6](#_Toc442802276)

[5. PROBLEM-BASED LEARNING APPROACH 7](#_Toc442802277)

[7. LEARNING OUTCOMES 8](#_Toc442802278)

[7.1 KNOWLEDGE 8](#_Toc442802279)

[7.2 SKILLS 8](#_Toc442802282)

[7.3 COMPETENCE 8](#_Toc442802284)

[8. ASSESSMENT 9](#_Toc442802285)

[8.1 Individual Assignments 9](#_Toc442802286)

[8.2 Group Term Paper – Research Proposal 9](#_Toc442802287)

[9. RECOMMENDED TEXT 9](#_Toc442802288)

[DETAILED CLASS SCHEDULE 10](#_Toc442802289)

[10. SCHEDULE OF SESSIONS 10](#_Toc442802290)

[11. SESSION 1 – Computer Science and its Applications 10](#_Toc442802291)

[11.1 Overview 10](#_Toc442802292)

[11.2 Goals and Objectives 10](#_Toc442802293)

[11.3 Activities and Assignments 11](#_Toc442802294)

[12. SESSION 2 – Information Representation 11](#_Toc442802295)

[12.1 Overview 11](#_Toc442802296)

[12.2 Goals and Objectives 11](#_Toc442802297)

[12.3 Activities and Assignments 11](#_Toc442802298)

[13. SESSION 3 – Communications and Internet Technologies 12](#_Toc442802299)

[13.1 Overview 12](#_Toc442802300)

[13.2 Goals and Objectives 12](#_Toc442802301)

[13.3 Activities and Assignments 13](#_Toc442802302)

[14. SESSION 4 – Computer Systems Fundamentals (Hardware) 13](#_Toc442802303)

[14.1 Overview 13](#_Toc442802304)

[14.2 Goals and Objectives 13](#_Toc442802305)

[14.3 Activities and Assignments 14](#_Toc442802306)

[15. SESSION 5 – Computer Systems Fundamentals (Software) 14](#_Toc442802307)

[15.1 Overview 14](#_Toc442802308)

[15.2 Goals and Objectives 14](#_Toc442802309)

[15.3 Activities and Assignments 15](#_Toc442802310)

[16. SESSION 6 – Security and Ethics 15](#_Toc442802311)

[16.1 Overview 15](#_Toc442802312)

[16.2 Goals and Objectives 15](#_Toc442802314)

[16.3 Activities and Assignments 16](#_Toc442802315)

[17. SESSION 7 – Databases and Data Modelling 16](#_Toc442802316)

[17.1 Overview 16](#_Toc442802317)

[17.2 Goals and Objectives 16](#_Toc442802319)

[17.3 Activities and Assignments 17](#_Toc442802320)

[18. SESSION 8 – Algorithmic Thinking 17](#_Toc442802321)

[18.1 Overview 17](#_Toc442802322)

[18.2 Goals and Objectives 17](#_Toc442802324)

[18.3 Activities and Assignments 18](#_Toc442802325)

[19. SESSION 9 – Data Types 18](#_Toc442802326)

[19.1 Overview 18](#_Toc442802327)

[19.2 Goals and Objectives 18](#_Toc442802329)

[19.3 Activities and Assignments 19](#_Toc442802330)

[20. SESSION 10 – Fundamentals of Programming 19](#_Toc442802331)

[20.1 Overview 19](#_Toc442802332)

[20.2 Goals and Objectives 19](#_Toc442802334)

[20.3 Activities and Assignments 19](#_Toc442802335)

[21. SESSION 11 – Software Development 19](#_Toc442802336)

[21.1 Overview 19](#_Toc442802337)

[21.2 Goals and Objectives 19](#_Toc442802338)

[21.3 Activities and Assignments 20](#_Toc442802339)

[22. SESSION 12 – Simulation and Systematic Approach to Problem Solving 20](#_Toc442802340)

[22.1 Overview 20](#_Toc442802341)

[22.2 Goals and Objectives 20](#_Toc442802342)

[22.3 Activities and Assignments 20](#_Toc442802343)

[23. SESSION 13 – AI and the Limits of Computing 20](#_Toc442802344)

[23.1 Overview 20](#_Toc442802345)

[23.2 Goals and Objectives 20](#_Toc442802346)

[23.3 Activities and Assignments 20](#_Toc442802347)

[APPENDIX A 21](#_Toc442802348)

COURSE DESCRIPTION

# COURSE WEBSITE

http://sakai.ug.edu.gh/XXXXXXXXXXX

# INSTRUCTORS

|  |  |
| --- | --- |
| **Prof. F.A. Katsriku**  Dept. of Computer Science  School of Physical and Mathematical Sciences  P. O. Box LG 163  Legon, Accra  Email:  [fkatsriku@gmail.com](mailto:fkatsriku@gmail.com)  Email: [fkatsriku@ug.edu.gh](mailto:fkatsriku@ug.edu.gh)  Office: RM 02, RIPS Building |  |

# Support Contact Information

Department of Computer Science

University of Ghana

P. O. Box LG163

Legon, Accra

Phone: XXXXXXXXX

Email: XXXXXXXXXXXX

1. **OVERVIEW**

This course is designed to give you a first introduction to the exciting field of computer science and computing. It is hoped that you will use the skills and knowledge of computer science acquired through this course in one of

three ways:

• to provide a general understanding and perspective of the development of computer technology and

systems, which will inform your decisions and support your participation in an increasingly technologically

dependent society

• to provide the necessary skills and knowledge that will lay the foundations for the use of computer science in any specialist area.

• to develop your knowledge and understanding of computer science by seeking to pursue the program at an advance level in higher education, where this qualification will provide a useful foundation for further study of computer science or more specialist aspects of computer science.

**Course Objectives**: Computer science is the study of the foundational principles and practices of computation and computational thinking and their application in the design and development of computer systems.

This course aims to encourage the development of computational thinking, that is thinking about what can

be computed and how by the use of abstraction and decomposition. It includes consideration of the data

required. Learning computational thinking involves learning to program, by writing computer code, because

this is the means by which computational thinking is expressed.

The aims of this course in Computer Science are:

• to develop computational thinking

• to develop an understanding of the main principles of solving problems using computers

• to develop an understanding that every computer system is made up of subsystems, which in turn

consist of further subsystems

• to develop an understanding of the component parts of computer systems and how they interrelate,

including software, data, hardware, communications and people

• to acquire the skills necessary to apply this understanding to develop computer-based solutions to

problems.

# PROBLEM-BASED LEARNING APPROACH

Problem-based learning (PBL) is a student-centered pedagogy in which students learn about a subject through the experience of problem solving. The goals of PBL are to help the students develop flexible knowledge, effective problem solving skills, self-directed learning, effective collaboration skills and intrinsic motivation. This course will use a problem-based learning approach.

Working in groups, students identify what they already know, what they need to know, and how and where to access new information that may lead to resolution of the problem. The role of the instructor/lecturer/tutor is to facilitate learning by supporting, guiding, and monitoring the learning process. The tutor will help build students' confidence to take on the problem, and encourage the students, while also stretching their understanding.

1. **COURSE FORMAT**

The course content will be delivered online through a series of short video lectures and will be managed using the SAKAI Learning Management System (Sakai LMS). The Sakai LMS will be used to deliver the following content

* Session Slides
* Video Recordings
* Session Reading Materials
* Assessments – Tests, Quizzes and Assignments (including a Problem-based Term Paper)
* Group activities – Discussions and Presentations
* Chat room discussions

Announcements will be posted to the course website and/email accordingly. It is the responsibility of students to check on announcements made in class, on the Course Website, and through email.

# LEARNING OUTCOMES

The learning outcomes for the course are outlined along three strands: knowledge, skills and outcomes.

## KNOWLEDGE

Students must have knowledge on

1. show understanding of the characteristics and methods of operation of component parts of computer

systems (hardware, software, communication) and their subsystems

1. describe, explain and use various different methods of representing data for use in computer systems
2. comment critically on ethical issues arising from the use of computer solutions.

## SKILLS

Students must be able to

1. apply knowledge with understanding to computational problems
2. select, justify and apply appropriate techniques and principles to develop data structures and algorithms

for the solutions of computational problems

1. design, implement, document and evaluate an effective solution using appropriate hardware, software and programming languages.

## COMPETENCE

Students should in the future be able to

1. develop independent thinking skills
2. apply knowledge and understanding to new as well as familiar situations
3. handle and evaluate different types of information sources
4. think logically and present ordered and coherent arguments
5. make judgements, recommendations and decisions
6. present reasoned explanations, understand implications and communicate them clearly and logically
7. work and communicate in English.

# ASSESSMENT

The assessment for this course has been designed to help all students to maximize their individual and group/team learning opportunities. A summary of the assessment tasks is provided below.

|  |  |  |  |
| --- | --- | --- | --- |
| **ITEM** | **FORM OF ASSESSMENT** | **DELIVERED** | **MARKS** |
| Individual Assignments | Multiple Choice Questions (MCQs)  Exercises  Presentation/class work  Mid semester exam | Alongside Sessions | 20%  20% |
| Group Assignments | Term Project/Paper/ Presentation | Beginning of Semester | 20% |
| Written Examination | Semester Examination | End of semester | 40% |
| **Total** | | | 100% |

**Criteria for passing**

You must pass both the coursework and the written examination to pass the course.

## Individual Assignments

Individual assignments will be provided at end of sessions. These assignments may be in the form of Multiple Choice Questions (MCQs), Short Essays and/or a Term paper or presentation. Deadlines will be provided for each assignment with respect to scheduling of the sessions.

## Group Term Paper – Research Proposal

Use the following outline provided in Appendix A to complete both Individual and group assignments

# RECOMMENDED TEXT

Heathcote, P. and Bond, K. (2013). *A Level Computing.* London: BPP (Letts Educational)

ISBN 1 85758 601 8, 244pages.

Brookshear, J. G. (2011). *Computer Science: An Overview* (11th ed). Boston: Addison Wesley

ISBN-10: 0132569035, ISBN-13: 978-0132569033, 624 pages

Geoffrey, S. (2008). *Introduction to Computer Information System* (2nd ed.). Iowa: Kendall

Hunt Publishing. ISBN-10: 075754911X, ISBN-13: 978-0757549113, 650 pages

Kamaljeet, S.(2013). *Fundamentals of Computing* (2nd ed.). Iowa: Kendall Hunt Publishing.

ISBN-10: 0757584764, ISBN-13: 978-0757584763, 746 pages

Miller, M. (2011). *Absolute Beginner’s Guide to Computer Basics* (4th ed.). New York: Que

Publishing. ISBN-10: 078973673X, ISBN-13: 978-0789736734, 456 pages

DETAILED CLASS SCHEDULE

The course is organized into 13 SESSIONS along the following lines: (1) Overview; (2) Goals and Objectives; and (3) Activities and Assignments.

# SCHEDULE OF SESSIONS

|  |  |
| --- | --- |
| **Week** | **Session** |
| 1 | Sessions 1 |
| 2 | Session 2 |
| 3 | Sessions 3 |
| 4 | Sessions 4 |
| 5 | Session 5 |
| 6 | Session 6 |
| 7 | Session 7 |
| 8 | Session 8 |
| 9 | Session 9 |
| 10 | Session 10 |
| 11 | Session 11 |
| 12 | Session 12 |
| 13 | Session 13 |

# SESSION 1 – Computer Science and its Applications

## Overview

* Computers have permeated all areas of our lives. The range of computer applications is vast, covering all areas mentioned in this course as well as many more and you will need to develop a general knowledge and awareness of a variety of application areas. The best way of doing this is to read widely, newspapers, relevant magazines and textbooks, and to watch documentary programs on TV. For each application area you should be able to explain
* Purpose of the application
* Overall system design
* How data is captured and input to the computer
* What hardware is used
* What provision could be made for system failure
* The effect of the computer system on individuals in organisations and on society in general

The lecture starts with a brief overview of the history of computer science before considering the applications areas.

## Goals and Objectives

At the end of the session, the student will

* Describe how computers are used in a variety of contexts
* Describe the main features and capabilities of CAD/CAM systems, robots and expert systems
* Describe the main features and use of common application packages such as word-processing, desktop publishing, spreadsheets and graphics packages
* Distinguish between bitmapped and vector graphics
* Describe the benefits and drawbacks of computerization to an organisation, its employees and members of the public
* Discuss the effects of computerisation from a social, legal or economic point of view
* Describe the main clauses of the data protection act and copyright laws
* Describe in general terms the historical development of computer science

## Activities and Assignments

This week complete the following tasks:

Your weekly assignment will be given in class.

# SESSION 2 – INFORMATION REPRESENTATION

## Overview

Unlike human beings’ computers represent information in an entirely different manner. This lecture is to help you understand and appreciate how different forms of information are represented in the computer. The focus will be on the fundamental types of information including number, image, sound and video. We will also discuss how this information can be efficiently represented using data compression techniques.

## Goals and Objectives

At the end of the session, the student will

**Number representation**

• show understanding of the basis of different number systems and use the binary, denary and

hexadecimal number system

• convert a number from one number system to another

• express a positive or negative integer in two’s complement form

• show understanding of, and be able to represent, character data in its internal binary form

depending on the character set used (Candidates will not be expected to memorise any particular

character codes but must be familiar with ASCII and Unicode.)

• express a denary number in Binary Coded Decimal (BCD) and vice versa

**Images**

• show understanding of how data for a bitmapped image is encoded

• use the terminology associated with bitmaps: pixel, fi le header, image resolution, screen

resolution

• perform calculations estimating the fi le size for bitmapped images of different resolutions

• show understanding of how data for a vector graphic is represented and encoded

• use the terminology associated with vector graphics: drawing object, property and drawing list

• show understanding of how typical features found in bitmapped and vector graphics software are

used in practice

• justify where bitmapped graphics and/or vector graphics are appropriate for a given task

**Sound**

• show understanding of how sound is represented and encoded

• use the associated terminology: sampling, sampling rate, sampling resolution

• show understanding of how fi le sizes depend on sampling rate and sampling resolution

• show understanding of how typical features found in sound editing software are used in practice

**Video**

• Show understanding of the characteristics of video streams:

– the frame rate (frames/second)

– interlaced and progressive encoding

– video interframe compression algorithms and spatial and temporal redundancy

– multimedia container formats

**Compression techniques**

• show understanding of how digital data can be compressed, using either ‘lossless’ (including runlength

encoding – RLE) or ‘lossy’ techniques

## Activities and Assignments

This week complete the following tasks:

Your weekly assignment will be given in class.

# SESSION 3 – COMMUNICATIONS AND INTERNET TECHNOLOGIES

## Overview

Computer communications are at the heart of the modern use of computers and it is important for you to understand how computer to computer communication is achieved. You will learn the basics of computer communications and how the internet works. You will learn to distinguish between serial and parallel communication, synchronous and asynchronous, distinguish bit rate and baud rate, the relation between bit rate and bandwidth. You will also learn about the fundamental concepts and terms used in networking, networking hosts, the internet and how it works as well as security on the internet.

## Goals and Objectives

At the end of the session, the student will

**Networks**

Understand the structure of the Internet.

Understand the role of packet switching and

routers and know the main components of a packet.

Define:

•• router

•• gateway.

Consider where and why they are used.

Explain how routing is achieved across the

Internet.

• explain the client-server model of networked computers

• give examples of applications which use the client-server model

• describe what is meant by the World Wide Web (WWW) and the Internet

• explain how hardware is used to support the Internet: networks, routers, gateways, servers

• explain how communication systems are used to support the Internet: The Public Switched

Telephone Network (PSTN), dedicated lines, cell phone network

• explain the benefits and drawbacks of using copper cable, fibre-optic cabling, radio waves,

microwaves, satellites

**IP addressing**

• explain the format of an IP address and how an IP address is associated with a device on a

network

• explain the difference between a public IP address and a private IP address and the implication for

security

• explain how a Uniform Resource Locator (URL) is used to locate a resource on the World Wide

Web (WWW) and the role of the Domain Name Service

**Client- and server-side scripting**

• describe the sequence of events executed by the client computer and web server when a web

page consisting only of HTML tags is requested and displayed by a browser

– Client-side

recognise and identify the purpose of some simple JavaScript code

describe the sequence of events executed by the client computer and web server when a

web page with embedded client-side code is requested and displayed by a browser

show understanding of the typical use of client-side code in the design of an application

– Server-side

recognise and identify the purpose of some simple PHP code

describe the sequence of events executed by the client computer and web server when a

web page with embedded server-side code is requested and displayed by a browser

show understanding that an appropriately designed web application for accessing

database data makes use of server-side scripting

Define serial and parallel transmission methods and discuss the advantages of serial over parallel transmission.

Define and compare synchronous and asynchronous data transmission.

Describe the purpose of start and stop bits in asynchronous data transmission

Understand:

•• physical star topology

•• logical bus network topology

and:

•• differentiate between them

•• explain their operation.

where they might be used:

•• peer-to-peer networking

•• client-server networking.

Explain symmetric and asymmetric (private/public key) encryption and key exchange.

Explain how digital certificates and digital signatures are obtained and used.

Discuss worms, trojans and viruses, and the vulnerabilities that they exploit.

Discuss how improved code quality, monitoring and protection can be used to address worms,

trojans and viruses.

## Activities and Assignments

This week complete the following tasks:

Your weekly assignment will be given in class.

# SESSION 4 – COMPUTER SYSTEMS FUNDAMENTALS (Hardware)

## Overview

Computers are fundamentally hardware equipment. The aim of this lecture is to acquaint you with the basic components of the computer. You will learn about the input output devices, main memory, secondary storage, logic gates, the architecture of the CPU, the fetch execute-cycle, assembly language and the instruction set of the computer.

## Goals and Objectives

At the end of the session, the student will

Input, output and storage devices

identify hardware devices used for input, output, secondary storage

show understanding of the basic internal operation of some types of I/O devices

show understanding of the need for secondary (including removable) storage

show understanding of the need for secondary (including removable) storage

**Main Memory**

show understanding of the need for primary storage

– explain the differences between RAM and ROM memory

– explain the differences between Static RAM (SRAM) and Dynamic RAM (DRAM)

**Logic gates and Logic circuits**

understand and define the functions of NOT, AND, OR, NAND, NOR and XOR (EOR) gates

including the binary output produced from all the possible binary inputs

• construct the truth table for each of the logic gates above

• construct a logic circuit from either:

– a problem statement

– a logic expression

• construct a truth table from either:

– a logic circuit

– a logic expression

• show understanding that some circuits can be constructed with fewer gates to produce the same

Outputs

**CPU Architecture**

show understanding of the basic Von Neumann model for a computer system and the stored

program concept

• show understanding of the roles carried out by registers, including the difference between

general purpose and special purpose registers: Program Counter, Memory Data Register,

Memory Address Register, Index Register, Current Instruction Register and Status Register

• show understanding of the roles carried out by the Arithmetic and Logic Unit (ALU), Control Unit

and system clock

• show understanding of how data are transferred between various components of the computer

system using the address bus, data bus and control bus

• show understanding of how the bus width and clock speed are factors that contribute to the

performance of the computer system

• show understanding of the need for ports, for example Universal Serial Bus (USB), to provide the

connection to peripheral devices

**The fetch-execute cycle**

• describe the stages of the fetch-execute cycle

• show understanding of ‘register transfer’ notation

• describe how interrupts are handled

**The processor’s instruction set**

• show understanding that the set of instructions are grouped into instructions for:

– data movement (register to main memory and vice versa)

– input and output of data

– arithmetic operations

– unconditional and conditional jump instructions

– compare instructions

– modes of addressing: immediate, direct, indirect, indexed, relative

**Assembly language**

• show understanding of the relationship between assembly language and machine code, including

symbolic and absolute addressing, directives and macros

• describe the different stages of the assembly process for a ‘two-pass’ assembler for a given

simple assembly language program

• trace a given simple assembly language program

## Activities and Assignments

This week complete the following tasks:

Your weekly assignment will be given in class.

# SESSION 5 – COMPUTER SYSTEMS FUNDAMENTALS (Software)

## Overview

The basic computer equipment or hardware cannot do much without the programs or software. The major advances we witness in today’s world are down to developments in computer software. This lecture will introduce you to what constitutes computer software. You will learn about operating systems, utility programs, library programs and language translators.

## Goals and Objectives

At the end of the session, the student will

**Operating system**

• describe why a computer system requires an operating system

• explain the key management tasks carried out by the operating system

**Utility programs**

• show an understanding of the need for typical utility software used by a PC computer system:

– disk formatter

– virus checker

– defragmenter software

– disk contents analysis/disk repair software

– file compression

– backup software

**Library programs**

• show an understanding that software under development is often constructed using existing code

from program libraries

• describe the benefits to the developer of software constructed using library fi les, including

Dynamic Link Library (DLL) fi les

• draw on experience of the writing of programs which include library routines

**Language translators**

• show an understanding of the need for:

– assembler software for the translation of an assembly language program

– a compiler for the translation of a high-level language program

– an interpreter for execution of a high-level language program

• explain the benefits and drawbacks of using either a compiler or interpreter

• show awareness that high-level language programs may be partially compiled and partially

interpreted, such as Java

## Activities and Assignments

This week complete the following tasks:

Your weekly assignment will be given in class.

# SESSION 6 – COMPUTER SECURITY AND ETHICS

## Overview

## This session surveys the history and examples of computer crimes, their types, costs on society, and strategies of detection and prevention. The session also describes the types of ethical decisions that IT professionals must make, as well as the business needs they must balance when dealing with security issues. In addition to providing a useful classification of computer crimes and their perpetrators, the session explains both how to implement trustworthy computing to manage security vulnerabilities and how to respond to specific security incidents to fix problems quickly and improve ongoing security measures.

## Goals and Objectives

At the end of the session, the student will

**Data security**

• explain the difference between the terms security, privacy and integrity of data

• show appreciation of the need for both the security of data and the security of the computer

system

• describe security measures designed to protect computer systems, ranging from the stand-alone

PC to a network of computers, including:

– user accounts

– firewalls

– general authentication techniques, including the use of passwords and digital signatures

• describe security measures designed to protect the security of data, including:

– data backup

– a disk-mirroring strategy

– encryption

– access rights to data (authorisation)

• show awareness of what kind of errors can occur and what can be done about them

**Data integrity**

• describe error detection and correction measures designed to protect the integrity of data,

including:

– data validation

– data verifi cation for data entry

– data verifi cation during data transfer, including

○ parity check

○ checksum check

**Ethics and the computing professional**

• show a basic understanding of ethics

• explain how ethics may impact on the job role of the computing professional

• show understanding of the eight principles listed in the ACM/IEEE Software Engineering Code of

Ethics

• demonstrate the relevance of these principles to some typical software developer workplace

scenarios

• show understanding of the need for a professional code of conduct for a computer system developer

**Ownership of software and data**

• show understanding of the concept of ownership and copyright

• describe the need for legislation to protect ownership, usage and copyright

• discuss measures to restrict access to data made available through the Internet and World Wide

Web

• show understanding of the implications of different types of software licensing: Free Software

Foundation, the Open Source Initiative, shareware and commercial software

## Activities and Assignments

This week complete the following tasks:

Your weekly assignment will be given in class.

# SESSION 7 – DATABASES AND DATA MODELING

## Overview

## An important use of computers is in the management of data. Data if properly managed can produce a wealth of information to aid businesses in their decision-making process. In this lecture, you will learn about tools that have been developed for the management of data. We will discuss database management systems, relational database modeling and the use of data definition and data manipulation languages.

## Goals and Objectives

At the end of the session, the student will

**Database Management Systems (DBMS)**

• show understanding of the limitations of using a file-based approach for the storage and retrieval

of data

• describe the features of a relational database which address the limitations of a file-based

approach

• show understanding of the features provided by a DBMS to address the issues of:

– data management, including maintaining a data dictionary

– data modelling

– logical schema

– data integrity

– data security, including backup procedures and the use of access rights to individuals/groups

of users

• show understanding of how software tools found within a DBMS are used in practice:

– developer interface

– query processor

• show awareness that high-level languages provide accessing facilities for data stored in a

database

**Relational database modelling**

• show understanding of, and use, the terminology associated with a relational database model:

entity, table, tuple, attribute, primary key, candidate key, foreign key, relationship, referential

integrity, secondary key and indexing

• produce a relational design from a given description of a system

• use an entity-relationship diagram to document a database design

• show understanding of the normalisation process: First (1NF), Second (2NF) and Third Normal

Form (3NF)

• explain why a given set of database tables are, or are not, in 3NF

• make the changes to a given set of tables which are not in 3NF to produce a solution in 3NF, and

justify the changes made

**Data Definition Language (DDL) and Data Manipulation Language (DML)**

• show understanding that DBMS software carries out:

– all creation/modification of the database structure using its DDL

– query and maintenance of data using its DML

• show understanding that the industry standard for both DDL and DML is Structured Query

Language (SQL)

– show understanding of a given SQL script

– write simple SQL (DDL) commands using a sub-set of commands for:

○ creating a database (CREATE DATABASE)

○ creating a table definition (CREATE TABLE)

○ changing a table definition (ALTER TABLE)

○ adding a primary key or foreign key to a table (ADD PRIMARY KEY)

– write a SQL script for querying or modifying data (DML) which are stored in (at most two)

database tables

○ Queries:

○ SELECT, FROM, WHERE, ORDER BY, GROUP BY, INNER JOIN

– Data maintenance:

○ INSERT INTO, DELETE FROM, UPDATE

## Activities and Assignments

This week complete the following tasks:

Your weekly assignment will be given in class

# SESSION 8 – ALGORITHMIC THINKING

## Overview

## The computer without human instructions cannot do anything on its own. Human beings develop those instructions called programs that cause the computer to behave in the desired manner. This calls for clear and precise instructions since the computer is devoid of intuition and cannot second guess what you are thinking or have in mind. This lecture will introduce you to how problems can be solved using the computer. You will learn to write probably your first program. We will discuss algorithms and those features that make for a good algorithm and provide examples of some algorithms.

## Goals and Objectives

At the end of the session, the student will

**Algorithms**

• show understanding that an algorithm is a solution to a problem expressed as a sequence of

defined steps

• use suitable identifier names for the representation of data used by a problem

– summarise identifier names using an identifier table

• show understanding that many algorithms are expressed using the four basic constructs of

assignment, sequence, selection and repetition

• show understanding that simple algorithms consist of input, process, output at various stages

• document a simple algorithm using:

– structured English

– pseudocode

– program flowchart

• derive pseudocode or a program flowchart from a structured English description of a problem

• derive pseudocode from a given program flowchart or vice versa

• use the process of stepwise refinement to express an algorithm to a level of detail from which

the task may be programmed

• decompose a problem into sub-tasks leading to the concept of a program module (procedure/

function)

• show an appreciation of why logic statements are used to define parts of an algorithm solution

• use logic statements to define parts of an algorithm solution

**Structure chart**

• use a structure chart to express the parameters passed between the various modules/

procedures/functions which are part of the algorithm design

• describe the purpose of a structure chart

• construct a structure chart for a given problem

• derive equivalent pseudocode from a structure chart

**Corrective maintenance**

• perform white-box testing by:

– selecting suitable data

– using a trace table

• identify any error(s) in the algorithm by using the completed trace table

• amend the algorithm if required

**Adaptive maintenance**

• make amendments to an algorithm and data structure in response to specification changes

• analyse an existing program and make amendments to enhance functionality

## Activities and Assignments

This week complete the following tasks:

Your weekly assignment will be given in class

# SESSION 9 – Fundamentals of Programming I

## Overview

## In this lecture you will learn how to write simple programs in a high-level language. You will also learn about data types including arrays and files.

## Goals and Objectives

At the end of the session, the student will

**Data types**

• select appropriate data types for a problem solution

• use in practical programming the data types that are common to procedural high-level languages:

integer, real, char, string, Boolean, date (pseudocode will use the following data types:

INTEGER, REAL, CHAR, STRING, BOOLEAN, DATE, ARRAY, FILE )

• show understanding of how character and string data are represented by software including the

ASCII and Unicode character sets

**Arrays**

• use the technical terms associated with arrays including upper and lower bound

• select a suitable data structure (1D or 2D array) to use for a given task

• use pseudocode for 1D and 2D arrays (pseudocode will use square brackets to contain the array

subscript, for example a 1D array as A[1:n] and a 2D array as C[1:m, 1:n] )

• write program code using 1D and 2D arrays

• write algorithms/program code to process array data including:

– sorting using a bubble sort

– searching using a linear search

**Files**

• show understanding of why files are needed

• use pseudocode for file handling

**Programming basics**

• write a program in a high-level language

• implement and write a program from a given design presented as either a program flowchart or

pseudocode

• write program statements for:

– the declaration of variables and constants

– the assignment of values to variables and constants

– expressions involving any of the arithmetic or logical operators

– input from the keyboard and output to the console

**Transferable skills**

• recognise the basic control structures in a high-level language other than the one chosen to be

studied in depth

• appreciate that program coding is a transferable skill

**Selection**

**Use of CASE structure**

**Iteration Built-in functions**

**Structure programming**

## Activities and Assignments

This week complete the following tasks:

Your weekly assignment will be given in class

# SESSION 10 – Fundamentals of Programming II

## Overview

## This will be a continuation of the previous lecture

## Goals and Objectives

At the end of the session, the student will

1. See session 9

## Activities and Assignments

This week complete the following tasks:

Your weekly assignment will be given in class

# SESSION 11 – SOFTWARE DEVELOPMENT

## Overview

## Software development can be a complex process. In fact, most software today are very complex and there must be systematic way of developing software. This lecture aims to acquaint you with some of the methods used in the development of software. You will learn about programming, program testing and testing strategies.

## Goals and Objectives

At the end of the session, the student will

**Programming**

• show understanding of the design, coding and testing stages in the program development cycle

• show understanding of how to write, translate, test and run a high-level language program

• describe features found in a typical Integrated Development Environment (IDE):

– for coding, including context-sensitive prompts

– for initial error detection, including dynamic syntax checks

– for presentation, including prettyprint, expand and collapse code blocks

– for debugging, including: single stepping, breakpoints, variables/expressions report window

**Program testing**

• show understanding of ways of exposing faults in programs and ways of avoiding faults

• locate and identify the different types of errors:

– syntax errors

– logic errors

– run-time errors

• correct identified errors

**Testing strategies**

• choose suitable data for black-box testing

• choose suitable data for white-box testing

• understand the need for stub testing

## Activities and Assignments

This week complete the following tasks:

Your weekly assignment will be given in class

# SESSION 12 – SIMULATION AND SYSTEMATIC APPROACH TO PROBLEM SOLVING

## Overview

When it is required to understand the functioning of a system very often this done by modeling and simulating the system on a computer. This lecture will introduce you to the world of computer simulations, how simulations are categorized and we will then look at some examples

## Goals and Objectives

At the end of the session, the student will

1. Categorize simulations of various types
2. Define problems and develop simulations to understand the problem

## Activities and Assignments

This week complete the following tasks:

Your weekly assignment will be given in class

# SESSION 13 – AI AND THE LIMITS OF COMPUTING

## Overview

Artificial Intelligence deals with the branch of computer science that studies the use of

computers to perform computational processes normally. Increasingly computers are being made to take over many of the roles of humans. This lecture will discuss the fascinating field of computer science and how far it can go. You learn about the various areas of AI including expert systems, machine learning, knowledge representation, robotics and natural language processing. Finally, we will touch on the problem of intractability and unsolvable problems.

## Goals and Objectives

At the end of the session, the student will learn about

Expert systems

Machine learning

Knowledge representation

Robotics

Natural Language processing

Intractability

## Activities and Assignments

This week complete the following tasks:

Your weekly assignment will be given in class

APPENDIX A

**GROUP ASSIGNMENT**

**Submission Guidelines:** Each assignment should be presented with a cover page which should state out the title of the assignment and also outline the name, index number and email address of students in the group. Assignment should be single-spaced, font-size 12, Times New Roman. Submit online through Sakai LMS by the end of the 12th Week.

Each assignment, therefore, should be carefully edited for grammar, typos, and/or spelling mistakes. Appropriate citations when applicable should be provided using the American Psychological Association (APA) format. The following website offers information about writing and formatting papers in the APA style including general format, reference of the works of others in your texts, reference list, examples, notes, and additional resources:

* http://owl.english.purdue.edu/workshops/hypertext/apa/index.html

1. **Late Submission of Assignments**
   * + Any work submitted after the advertised deadline will attract a penalty in the form of reduction of marks. Up to 25% of marks available may be lost due to late submission