IT4104 - Programming II (Compulsory)

INTRODUCTION

This is one of the two compulsory courses designed for Semester 4 of the Bachelor of Information Technology Degree program. This course includes the essential components of data structures and algorithms, which can be used to manipulate data considering the computer's memory and it's intrinsic constraints. Candidate should be able to implement the learnt concepts using the Java programming language.

CREDITS: 04

LEARNING OUTCOMES

After successful completion of this course, students will be able to:

- Use common data structures used in applications
- Use common searching and sorting algorithms

MINOR MODIFICATIONS

When minor modifications are made to this syllabus, those will be reflected in the Virtual Learning Environment (VLE) and the latest version can be downloaded from the relevant course page of VLE. Please inform your suggestions and comments through the VLE. http://vle.bit.lk

ONLINE LEARNING MATERIALS AND ACTIVITIES

You can access all learning materials and this syllabus in the VLE: http://vle.bit.lk, if you are a registered student of BIT degree program. It is very important to participate in learning activities given in the VLE to learn this subject.

ONLINE ASSIGNMENTS

The assignments consist of two quizzes, assignment quiz 1 (It covers the first half of the syllabus) and assignment quiz 2 (It covers the second half of the syllabus). Maximum mark for a question is10, minimum mark for a question is 0 (irrespective of negative scores). Final mark is calculated considering 40% of assignment quiz 1 and 60% of assignment quiz 2. Pass mark for the online assignments in a course is 50. You are advised to do online assignments before the final exam of the course. It is compulsory to pass all online assignments to partially qualify to obtain year 2 certificate.

FINAL EXAMINATION

Final exam of the course will be held at the end of the semester.

Examination Paper will consist of two parts.

- Part 1 : 1 Hour paper consisting of Multiple Choice Questions
- Part 2 : 1 Hour paper consisting of Structured Questions

OUTLINE OF SYLLABUS

Topic	Hours
1- Overview of Data Structures	02
2- Stacks, Queues and Hashing	12
3- Linked Lists	05
4- Recursion	07
5- Trees	11
6- Graphs	10
7- Searching and Sorting Algorithms	13
Total for the subject	**60***

** Students may need more time to do relevant practical work.*

REQUIRED MATERIALS

Main Reading

Ref 1: Data Structures and Algorithms in Java by Adam Drozdek, Thomson learning, 2nd edition, 2006, ISBN: 81-315-0107-8.

Ref 2: Data Structures and Algorithms in Java by Robert Lafore, GC Join for Techmedia, 2nd edition, ISBN: 817635-186-5

DETAILED SYLLABUS:

Section 1 : Overview of Data Structures (02 hrs)

Instructional Objectives

- Describe the use of data structures
- List different implementation techniques
- Identify code segments of a data structure in the given example

Material /Sub Topics

1.1 Introduction to data structures [Ref 2 : pg. 1]

1.2 Practical data storage structures [Ref 2 : pg. 2]

1.3 Programmer's Tools for data storage [Ref 2 : pg. 3]

1.4 Real-world Modeling for data storage [Ref 2 : pg. 3]

Section 2 : Stacks, Queues and Hashing (12 hrs)

Instructional Objectives

- Describe use of stacks
- Describe the use of queues
- Compare different queue implementations
- List the features of different queues
- Explain hash functions

Material /Sub Topics

2.1 Stacks

2.1.1 Implementing a stack [Ref 2 : pg. 91 – 105]

2.1.2 Efficiency of stacks [Ref 2 : pg. 105]

2.2 Queues

2.2.1 Overview of queues [Ref 1 : pg. 149]

2.2.2 Queues and their different operations [Ref 1 : pg. 149]

2.2.3 Implementing queues - normal and circular methods [Ref 1 : pg. 150 -152]

2.2.4 Priority queues [Ref 1 : pg. 157 -163]

2.3 Hashing [Ref 1 : pg. 520 -522]

2.3.1 Hash Functions

Section 3 : Linked Lists (05 hrs)

Instructional Objectives

- Describe the use of Linked Lists
- Explain different implementation of Linked Lists

Material /Sub Topics

3.1 Single Linked Lists [Ref 1 : pg. 80]

3.2 Doubly Linked Lists [Ref 1 : pg. 95 – 99]

3.3 Circular Lists [Ref 1 : pg. 99 - 101]

3.4 Skip Lists [Ref 1 : pg. 101 - 107]

3.5 Self-Organizing Lists [Ref 1 : pg. 107 - 111]

Section 4 : Recursion (07 hrs)

Instructional Objectives

- Define recursive methods
- Describe method calls in a recursion
- List differences in implementing recursion

Material /Sub Topics

4.1 Definition of Recursion [Ref1 : pg. 169 - 172]

4.2 Recursion and methods [Ref1 : pg. 172 - 174]

4.3 How a recursion call is executed? [Ref1 : pg. 174 - 178]

4.4 The implementation of recursion

4.4.1 Tail Recursion [Ref1 : pg. 178 - 179]

4.4.2 Nontail Recursion [Ref1 : pg. 179 - 184]

4.4.3 Indirect Recursion [Ref1 : pg. 185 - 187]

4.4.4 Excessive Recursion [Ref1 : pg. 188 - 191]

Section 5 : Trees (11 hrs)

Instructional Objectives

- Define trees structure
- Describe the properties of trees, binary trees
- Describe the implementation of trees
- Describe and Implement tree traversal techniques
- Explain how a tree is balanced
- Describe the ways of adjusting a tree
- Explain the usage of a heap

Material /Sub Topics

5.1 Trees, Binary trees and Binary search trees [Ref 1 : pg. 214 - 218]

5.2 Implementation of Binary trees [Ref 1 : pg. 219 - 221]

5.3 Searching a Binary search tree [Ref 1 : pg. 221 - 223]

5.4 Ways of traversing a tree [Ref 1 : pg. 223 - 231]

5.4.1 Breadth-First Traversal

5.4.2 Depth-First Traversal

5.4.3 Stackless Depth-First Traversal

5.5 Insertion and deletion [Ref 1 : pg. 239 - 246]

5.6 Balancing a tree [Ref 1 : pg. 249 - 260]

5.6.1 The DSW Algorithm

5.6.2 AVL Trees

5.7 Self-Adjusting trees [Ref 1 : pg. 260 - 267]

5.8 Heaps [Ref 1 : pg. 267 - 272]

Section 6 : Graphs (10 hrs)

Instructional Objectives

- Define the different types of graphs and their usage
- Implement the primary graph operations
- Describe and implement the Graph Traversal techniques, Connectivity and determination of Shortest Path

Material /Sub Topics

6.1 Definition of different Graphs [Ref 1 : pg. 376 - 377]

6.2 Graph Representation [Ref 1 : pg. 377 - 379]

6.3 Graph Traversals [Ref 1 : pg. 379 - 382]

6.4 Shortest paths [Ref 1 : pg. 383 - 392]

6.5 Cycle Detection [Ref 1 : pg. 392 - 395]

6.6 Spanning Trees [Ref 1 : pg. 395 - 405]

6.7 Connectivity of graphs [Ref 1 : pg. 399 - 405]

6.8 Topological sort [Ref 1 : pg. 405 - 406]

6.9 Networks [Ref 1 : pg. 407 - 421]

Section 7 : Sorting and Searching Algorithms (13 hrs)

Instructional Objectives

- Define and describe selected searching and sorting Algorithms
- Implement selected searching and sorting Algorithms

Material /Sub Topics

7.1 Efficiency of Algorithms [Ref 1 : pg. 718 - 723]

7.1.1 Big O Notation

7.2 Searching algorithms [Ref 1 : pg. 221 – 223, Ref 2 : pg. 405 – 410, 503 - 518]

7.2.1 Binary search trees

7.2.2 B-trees

7.2.3 Breadth-First and Depth-First Search

7.2.4 Java Implementation

7.3 Sorting Algorithms [Ref 1 : pg. 470 - 501]

7.3.1 Insertion sort

7.3.2 Selection sort

7.3.3 Bubble sort

7.3.4 Shell sort

7.3.5 Merge sort

7.3.6 Quick sort

7.3.7 Heap sort

7.3.8 Radix sort

7.3.9 Java Implementation

PLATFORM

Any standard PC (Pentium) with a standard Java Compiler (JDK 1.5 or above).

Note: *Students are expected to use Java as the coding language for this module.*