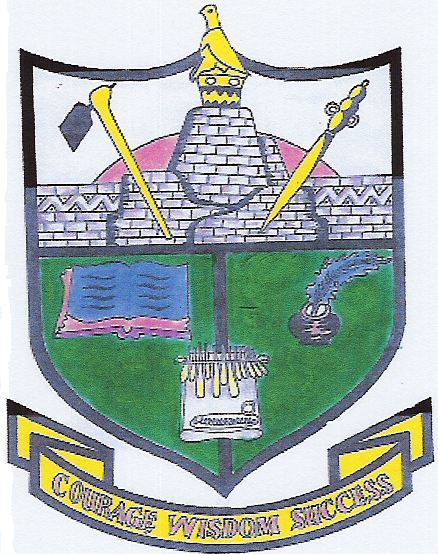
****

**Great Zimbabwe University**

**Gary Magadzire School of Agriculture and Natural Sciences**

**Course Outline:**

**Department of Mathematics and Computer Science**

**Subject Area: Computer Science**

**Course Name: Data Structures & Algorithms**

**Course Code: HCS201**

**Campus: GZU Main Campus**

**Semester 1, 2018**

**Time: 60 Hours**

**Contact Details for Instructors:**

|  |  |
| --- | --- |
| **Lecturer: Mr. A. Munthali**  **Location: Office #7 Garikai 1**  **Phone: 0735270920**  **Email: amunthali@gzu.ac.zw**  **Office Hours: 0800Hrs – 1600Hrs** | **Teaching Assistant:**  **Location:**  **Phone:**  **Email:**  **Office Hours:** |

[www.gzu.ac.zw](http://www.gzu.ac.zw)

PREAMBLE

The performance of a program on small inputs (typical in introductory computer science courses) gives no indication of how an algorithm will perform on the large program often found in real applications. In this course, we cover paper and pencil techniques for analyzing algorithms implementation. Timings of various algorithms, illustrate the practicality of the mathematical performances. Correctness is also an important issue, which we address. The techniques learned in the course are applied to well-studied classical problems including searching, sorting and some graph theory applications. The study of abstract data types is continued from HCS102 but the focus changes from that of understanding the data types to being able to make knowledgeable choices as the best data structures for a particular application.

OBJECTIVES

Students should be able to:

* Design and analyze algorithms.
* Implement some designed algorithms using selected data structures.
* Search and sort given elements using various searching and sorting methods.
* Compare various searching and sorting methods

COURSE CONTENT

1. INTRODUCTION TO DATA STRUCTURES

* Algorithms
* Sub routines
* Array Notation
* Analysis of algorithms
* Mathematical preliminaries
* Data type, data object and data structures
* Primitive data structures
* Records(structures)
* Files

1. ARRAYS

* Linear array notation
* Representation of linear Array in memory
* Traversing an Array
* Inserting an element
* Deleting an element
* Searching
* Multi Dimensional arrays

1. LINKED LISTS

* Representation of a list
* Allocation of memory nodes
* Garbage collection
* Advantages and disadvantages of Dynamic and Array implementation
* Operation of a list
* Reversing a list
* Sorting a list
* Double linked lists

1. STACKS

* Array representation of stack
* Linked list representation of stack
* Implementing two stacks in efficient manner
* Evaluation of postfix, prefix expression
* Prefix or polish notation
* Postfix or reverse polish notation
* Converting infix expression to post fix
* Checking parentheses

1. QUEUES

* Implementation
* Double ended queues
* Linked list representation of a queue
* Double linked representation of a D queue
* Priority queues

1. TREES

* Representation of tree in memory
* Binary tree
* Traversal of binary tree
* Traversal algorithms
* Conversion of general tree to binary tree

1. SEARCHING TECHNIQUES

* Linear search or sequential
* Binary search
* Hashing
* Hash tables
* Conflict resolution

1. INTERNAL SORTING TECHNIQUES

* Selection sort
* Bubble sort
* Insertion sort
* Merge sort
* Quick sort
* Heaps and heap sort
* Topological sorting
* Radix sort
* Shell sort

1. RECURSION

* factorial
* Fibonacci sequence
* recursion algorithm for binary tree traversal
* algorithm to find maximum value in an array
* finding maximum and minimum value
* binary search

METHOD OF EVALUATION

(Coursework 30%)

TWO ASSIGNMENTS (to cover both practical and theoretical concepts)

Quizzes

Presentations

In-class test.

A final three hour examination constituting 70% of the final mark.

NB it is the student’s responsibility to find out what he/she has missed if absent from lectures.

Non fulfillment of coursework disqualifies a student from sitting for the final examination.

Late submission of assignment attract a penalty commensurate with the number of days taken after deadline.

**Reference Textbooks**

1. **Algorithm design, foundations, analysis and internet examples**, M.T.Goodrich and R.Tamasia , John Wiley &Sons, inc
2. **Data structures & program design in C , 2nd Edition**: Robert Kruse,C.L.Tondo and Bruce Leong

Any other relevant text