AEN2205 Computer Programming (4 CU)

Lecturer: Mr. Hillary Kasedde (B.Sc. Mech. Eng., M.Sc.)
Ms.Fildah Ayaa (BSc. Agric Eng)

Course Type: CORE (B.Sc. Agric. Eng)

Course level: 2

Course Credits (CU): 4 CU i.e. 60 Contact Hours per semester

Course Duration: 15 weeks (45 hours) i.e. 45 LH, 30 PH

COURSE DESCRIPTION

Competency in a programming language is prerequisite to the study of computer engineering. Object-oriented programming, event-driven applications, and the use of extensive APIs (application programming interfaces) are fundamental tools that computer engineering students need early in their academic program.

2. COURSE OBJECTIVES

The objectives of this course are:

- To introduce the principles and fundamentals of computer programming.
- To equip the student with the skills of using a programming language to solve day to day problems.

Specific objectives:
On completion of this course the student should be able to:

- Describe how computer engineering uses or benefits from programming fundamentals.
- Identify the appropriate paradigm for a given programming problem.
- Use a suitable programming language to implement, test, and debug algorithms for solving simple problems.
- Describe the way a computer allocates and represents these data structures in memory.
- Outline the philosophy of object-oriented design and the concepts of encapsulation, sub classing, inheritance, and polymorphism.

.

3. RECOMMENDED REFERENCES FOR READING

- Paul J. Lucas ,The C ++ Programmer's Handbook, Prentice Hall 1994.
- Jean Ettinger,Programming in C++ Macmillan Press,2003
- Deitel and Deitel C++ How to program,4th Edition,Prentice Hall 2003.

4. COURSE CONTENT, METHODS OF INSTRUCTION, TOOLS AND EQUIPMENT REQUIRED

TOPIC	CONTENT	METHOD OF INSTRUCTION / Time allocated	TOOLS / EQUIPMENT NEEDED
Chapter 1. History and Overview	• reasons for studying programming fundamentals • define important programming	Interactive lectures (4 hrs)	Chalk / BB or Markers / Flip charts/LCD Projector/laptop

	areas • Contrast between an algorithm and a data structure • Distinguish between a variable, type, expression, and assignment • Highlight the role of algorithms in solving problems • Describe fundamental data structures	Tutorial (3 hrs)	
Chapter 2. Programming Languages	• Definition and History • Characteristics (Pragmatics, Semantics and Syntax) • Distinction between Text-based and Visual Programming • Classification (Categorical, Chronological and Generational) • Comparison of common programming languages (C, C++, C#, Java) • Programming errors and warnings (syntax, logical, etc.)	Interactive lectures (4 hrs) Tutorial (3 hrs)	Chalk / BB or Markers / Flip charts/LCD Projector/laptop /
Chapter 3. Programming Paradigms	• Definition and rationale of a programming paradigm • Types of programming paradigms e.g. Structured, Unstructured, Procedural, Object-oriented, Event-Drive, Generic etc. • Separation of behavior and implementation	Interactive lectures (8 hrs) Tutorial (3 hrs)	Chalk / BB or Markers / Flip charts/LCD Projector
Chapter 4. ISO/ANSI C++ Programming Fundamentals	• Bjarne Stroustrup Design rules • Console applications basics (Source file, Basic I/O, Standard I/O Consoles, Function main( )) • Fundamental data types • Expressions and operators • Control constructs (Conditional and Iterative) • Pointers and Named collections (Arrays, Enumerators, Bit-fields, Unions) • User-defined data types (Structures and Classes) • Functions (In-built and User-defined)	Interactive lectures (11 hrs) Practical (12 hrs)	Chalk / BB or Markers / Flip charts/LCD Projector/laptop / Computer lab

	• Object –oriented programming (Abstraction, Encapsulation, Inheritance, Composition, Polymorphism, Friend and Virtual Functions) • File I/O		
Algorithms and Problem-Solving	• Problem-solving strategies • The role of algorithms in the problem-solving process • Implementation strategies for algorithms • Debugging strategies • The concept and properties of algorithms • Structured decomposition	6 HOURS Tutorial (3 hrs)	Chalk / BB or Markers / Flip charts/LCD Projector/laptop / Computer lab
The Integrated Development Environment (IDE)	• Definition • Toolchains • Advantages of IDEs • Comparison of IDEs • Using a typical IDE (Visual Studio)	6 HOURS Practical(2 hrs)	Chalk / BB or Markers / Flip charts/LCD Projector/laptop / Computer lab
	• Evaluation	Tests (4 hrs)	Paper, printer, photocopier

5. SUMMARY OF TIME NEEDED

Interactive lectures covering theory	45 hrs
Computer-based practicals	26 hrs
Evaluation	4 hrs

6. OVERALL COURSE EVALUATION

Continuous Assessment	40%
Final examination	60%