

Course: CSC 112 – Introduction to Computer Science II (3 credits - Compulsory)
Course Duration: Three hours per week for 15weeks (30h (T) and 45h (P)), as taught in 2011/2012 academic session.

Lecturers: Ajiboye, A. R.

M.Sc. Computer Science (Ilorin), B.Sc. Computer Science (Ilorin), NCE (Oro), ANCS, MCPN
Department of Computer Science,
Faculty of Communication and Information Sciences,
University of Ilorin, Ilorin, Nigeria.

Email: ajibraheem@unilorin.edu.ng, ajibraheem@live.com

Office Location: Block 10, Room 10

Faculty of Communication and Information Sciences,
University of Ilorin, Ilorin, Nigeria.

Consultation Hours: 2.30pm – 3.30pm (Tuesdays and Thursdays)

Fagbemi, S. O.

B.Tech. Computer Science (Ogbomosho)

Department of Computer Science,
Faculty of Communication and Information sciences,
University of Ilorin, Ilorin, Nigeria.

E-mail: seun_fagbemi@yahoo.com

Office Location: Block 10, RoomF9

Faculty of Communication and Information Sciences,
University of Ilorin, Ilorin, Nigeria.

Consultation Hours: 11.00am – 12noon (Fridays)

Course Content

Operating Systems (As a computer operator); Low versus High Level language;
Characteristics of a High level language; Compiler versus Interpreter; Procedural versus Object-oriented Languages; Introduction to Programming (BASIC or FORTRAN); Problem solving, Flowcharts and Algorithms (Simple sorting and searching algorithms and some simple numerical analysis algorithms).

Course Description

This course provides a necessary foundation for Computer Science students. It begins by providing an introduction to operating systems as the basic software that controls the computer, its history, the current operating systems available and the future technologies. Fundamental computer concepts will be treated, including problem solving with algorithms and flowcharts. Sorting, searching, and simple numerical analysis algorithms are also treated. Furthermore, programming concept and tools are introduced; they include characteristics of programming languages, differences between high level and low-level languages, differences between compilers and interpreters, and procedural and object-oriented approaches to programming will also be considered. Finally, the BASIC programming language will be introduced.

Course Justification

The basic Computer concepts and programming paradigms taught in this course are fundamental to developing students as potential Computer Scientists. It is therefore necessary that students understand them fully.

Course Objectives

By the end of this course, students would be able to:

- (i) demonstrate an understanding of the operating system as a computer operator, describing its history and future technologies, its types, major functions and how it works;
- (ii) describe the history and application of programming languages, its function, the basic types and characteristics of each, and the differences between the major ones;
- (iii) juxtapose compiler with interpreter;
- (iv) identify the characteristics of and the differences between procedural and object-oriented languages and classify common programming languages based on these;
- (v) explain basic building blocks and stages of computer programming; and
- (vi) design computer programs in BASIC, using algorithms and flowcharts to develop codes to solve simple problems.

Course Requirements

Students are expected to attend 75% of the classes before they can sit for the examinations, they are also required to read widely making use of the reading list outlined in this courseware plus other relevant materials. Assignments are compulsory and they are to be submitted in word processed form. It is mandatory for every student to have a yahoo-mail account to facilitate contribution to the online discussion group.

Methods of Grading

S/N	Type of Grading	Score (%)
1.	Attendance	5
2.	Participation in class practical and online discussion group	5
3.	Assignments	10
4.	Test	10
5.	Final semester examination	70

Course Delivery Strategies

The traditional face-to-face method of delivering lectures will be employed for this course. Students will be distributed into groups in order to engage in practicals on various topics. An online discussion group would be available to every student so that ideas can be exchanged.

Practical Schedule

1. Acquaintance with the interface of common operating systems.
2. Introducing High level languages
3. Acquaintance with a compiler.
4. See example each of procedural and object-oriented languages
5. See a typical BASIC program
6. Run simple BASIC programs

LECTURE CONTENT

Weeks 1 and 2: Introduction and overview of Operating systems

Objectives

The objectives of the weeks' topic are to define what an operating system is, give its history, and describe its major functions. Also, it will be explained to the students how the operating system works and performs its functions. Types of operating systems will be considered alongside the components of modern operating systems. Examples of operating systems will be given and students will be acquainted with the interface of popular modern operating systems during the practical hour.

Description

First Hour(Week1)

Definitions of operating system

History of operating systems

Second Hour(Week 1)

Major functions of the operating system

Explanation of how the operating system works

First Hour (Week 2)

Types of Operating Systems

Operating System Components

Second Hour (Week 2)

Examples of popular modern operating systems

Brief description of popular operating systems

Practical Hours (Weeks 1 & 2)

Acquaintance with the interface of common operating systems.

Study Questions

1. Define operating system.
2. Give a brief history of operating systems.
3. Highlight the basic functions an effective operating system should perform.
4. Briefly explain each of the functions.
5. Briefly explain how the operating system works.
6. Enumerate the types of operating systems that we have.
7. Outline the system components that modern operating systems share.
8. List examples of modern popular operating systems.
9. How is a command-based operating system different from a graphical interface operating system?
10. What are the advantages of command-oriented operating systems over graphical interfaces operating systems?

Reading List

1. Tanenbaum³, A.S. (2001). *Modern Operating Systems* (pp. 1-20) (2nd Edn.)
2. Operating system – Wikipedia free encyclopedia³. Available at: http://en.wikipedia.org/wiki/operating_system. Accessed 19th September, 2011.
3. What is an operating system? – operatingSystems.com³. Available at http://www.operatingSystems.com/operating_systems. Accessed 19th September, 2011.
4. How operating systems works – HowStuffWorks.com³. Available at http://www.HowStuffworks.com/operating_systems. Accessed 19th September, 2011.

Week 3: An Overview of Computer Programming Languages

Objectives

The main objective of the week is to make the students understand what a programming language is, carefully examining the types of programming language, the features of each and the differences among them. Also, students will be equipped with the knowledge of language structure and components.

Description

First Hour

Introduction and history of programming languages

Types of programming languages

Second Hour

General features and classification of High level languages

Examples of High level languages

Definition and features of Machine and Low level languages

Differences between High and Low level languages

Practical Hour

Introducing High level languages

Study Questions

1. What is a programming language?
2. Give a brief history of programming languages.
3. Name the various types of programming languages, highlighting the features of each.
4. Give examples of high and low level languages.
5. Differentiate between high and low level languages.
6. How can programming languages be compared?

Reading List

1. French^{1,2}, C.S. (2001). *Introduction to Computer Science* (pp. 339-359). (5th Edn.). London: BookPower. ISBN 0-8264-6761-X.
2. High-level programming language – Wikipedia free encyclopedia³. Available at: <http://en.wikipedia.org/wiki/High-level> programming language. Accessed 19th September, 2011.
3. High Level vs. Low level Languages – ctc-control.com³. Available at <http://www.ctc-control.com/High Level vs. Low Level languages>. Accessed 19th September, 2011.

Week 4: Translators - Compilers and Interpreters

Objectives

This week's topic deals with translators, how they are used, and how they work. At the end of the week, the students will be able to identify the characteristics and basic functions of a compiler and an interpreter and differentiate between the two.

Description

First Hour

Introduction to translators.

Definition of a compiler.

Characteristics and functions of a compiler.

Second Hour

Definition of an interpreter.

Characteristics and functions of an interpreter.

Differences between a compiler and an interpreter.

Practical Hour

Acquaintance with a compiler.

Study Questions

1. What is a compiler?
2. Highlight the characteristics and functions of a compiler.
3. What is an interpreter?
4. List the characteristics and functions of an interpreter.
5. Identify the differences between a compiler and an interpreter.

Reading List

1. French^{1,2}, C.S. (2001). *Introduction to Computer Science* (pp. 393-401). (5thEdn.). London: BookPower. ISBN 0-8264-6761-X.
2. Compiler – Wikipedia free encyclopedia³. Available at: <http://en.wikipedia.org/wiki/compiler>. Accessed 19th September, 2011.

Week 5: Procedural and Object-oriented languages

Objectives

The main objectives of this week are to explain to students what procedural and object-oriented programming languages are, and describing the differences between the two, giving examples.

Description

First Hour

Procedural language

Object-oriented language

Second Hour

Major differences between procedural and object-oriented languages

Practical Hour

See example each of procedural and object-oriented languages

Study Questions

1. What is a procedural programming language?
2. Give some features of a procedural language.
3. What is an object-oriented language?
4. Describe the features of an object-oriented language.
5. Differentiate between a procedural and object-oriented language.

Reading List

1. French^{1, 2}, C.S. (2001). *Introduction to Computer Science* (pp. 391-392). (5thEdn.). London: BookPower. ISBN 0-8264-6761-X.
2. What is the difference between object-oriented and procedural languages? – Wikianswers³. Available at: [http://en.answers.com/What is the difference between object-oriented and procedural programming languages?](http://en.answers.com/What_is_the_difference_between_object-oriented_and_procedural_programming_languages?) Accessed 20th September, 2011.

Weeks 6: Problem solving - Introduction to Programming

Objectives

The objectives of this week are to explain to the students how to solve problems with the computer, introducing them to computer programming with emphasis on the building blocks and stages of programming.

Description

First Hour

Problem Solving with the Computer

Definition of a Computer Program

Characteristics of a good Computer program

Second Hour

The Programming process

Stages of Programming – Problem Definition, planning the solution, developing the prescribed method, coding, transcribing the instructions into machine sensible form, program debugging/testing, documentation.

Study Questions

1. Explain how the computer solves problems.
2. Define a computer program.
3. List the characteristics of a good computer program.
4. Outline the steps involved in programming.
5. Briefly explain each of the steps listed above.

Reading Lists

1. Walnum^{3, 4}, C. (2001). *Complete Idiot's Guide to Visual Basic 6* (pp. 14-21). Macmillan Computer Publishing. ISBN 078971812X.
2. French^{1, 2}, C.S. (2001). *Introduction to Computer Science* (pp. 204-206). (5thEdn.). London: BookPower. ISBN 0-8264-6761-X.

Weeks 7, 8 and 9: Algorithms and Flowcharts

Objectives

At the end of these weeks, students would have been equipped with the basic understanding of algorithms and flowcharts, with ability to demonstrate this understanding in solving simple problems. Also, simple sorting, searching and numerical analysis algorithms will be treated, examining how they work.

Description

First Hour (Week 7)

Definitions of an algorithm

Characteristics of an algorithm

Simple examples of algorithms

Second Hour (Week 7)

Definition and Importance of a flowchart

Flowchart Symbols

Guidelines for drawing flowcharts

Examples of flowcharts

First Hour (Week 8)

Solving problems with algorithms and flowcharts

Second Hour (Week 8)

Solving problems with algorithms and flowcharts

First Hour (Week 9)

Definitions of sorting and searching algorithms

Insertion Sort

Analysis of Insertion Sort

Binary Search

Second Hour (Week 9)

Numerical analysis algorithm

Study Questions

1. What is an algorithm?
2. Give the characteristics of an algorithm.
3. Define a flowchart.
4. Outline the most commonly used flowchart symbols and their functions.
5. What guidelines should be followed in drawing a flowchart?
6. What are the advantages of using algorithms and flowcharts in program development?
7. Suppose you are given twenty numbers, prepare the algorithm that adds up these numbers and find the average.
8. Draw the flowchart for the algorithm above.
9. Define a sorting algorithm.
10. List examples of sorting algorithms.
11. Explain what a searching algorithm means to you.
12. Give examples of searching algorithms.
13. What is a numerical analysis algorithm?
14. Give examples of numerical analysis algorithms.
15. Analyze the Insertion Sort algorithm.

Reading Lists

1. Cormen³, T. H., Leiserson, C.E., Rivest, R. L. & Stein, C. (2001). *Introduction to algorithms* (pp. 15-21). USA: MIT Press. ISBN 0-07-013151-1.
2. Knuth^{1,3}, D.E. (1998). *The Art of Computer Programming; Semi Numerical Algorithms* (pp. 80-105). California: Addison-Wesley. ISBN 0-201-03822-6
3. Insertion Sort – Wikipedia free encyclopedia³. Available at: http://en.wikipedia.org/wiki/Insertion_Sort. Accessed 19th September, 2011.

Week 10: Test

Objectives:

The main objectives of the test are to examine the ability of the students and to show how well they have understood the topics taught in the previous weeks.

Week 11: Introduction to programming in BASIC**Objectives**

Students will be introduced to the BASIC programming language, learning the elementary basic keywords, identifiers and the character set.

DescriptionFirst Hour

Elements of BASIC structure

Advantages of programming with BASIC

Structure of BASIC – rules on the structure of the language

Second Hour

BASIC Keywords

The BASIC character set

Practical Hour

See a typical BASIC program

Study Questions

1. What does the acronym 'BASIC' stand for?
2. What are the advantages of programming with BASIC?
3. Outline the rules on the structure of the language.
4. List some keywords in the BASIC programming language and explain what they mean.
5. Discuss the BASIC character set.

Reading List

1. French^{1, 2}, C.S. (2001). *Introduction to Computer Science* (pp. 390-391). (5th Edn.). London: BookPower. ISBN 0-8264-6761-X.
2. BASIC – Wikipedia free encyclopedia³. Available at: <http://en.wikipedia.org/wiki/BASIC>. Accessed 19th September, 2011.

Weeks 12, 13 and 14: BASIC Programming**Objectives**

The objective of this topic is to give students in-depth knowledge of BASIC programming

DescriptionFirst Hour (Week 12)

Variables

Constants – Numeric and String constant

Second Hour (Week 12)

Arithmetic Expressions – hierarchy of operations and rules guiding arithmetic expressions

Relational/Logical Expressions – relational/logical operators and their meanings

First Hour (Week 13)

Input-Output Statements – ‘LET’, ‘INPUT’, ‘READ’, ‘RESTORE’, ‘PRINT’, ‘OUTPUT’.

Second Hour (Week 13)

Program Control Statements – their syntax and use

Looping Statement

First Hour (Week 14)

Function and Subroutine

Second Hour (Week 14)

Library Functions in BASIC

Practical Hours (Weeks 12, 13 & 14)

Run simple BASIC programs

Study Questions

1. Define a variable.
2. Define a constant
3. Discuss the types of variables and constants allowed in BASIC.
4. What is an arithmetic expression?
5. List the hierarchy of operation in a BASIC arithmetic expression.
6. What rules apply to an arithmetic expression in BASIC?
7. What is a relational expression?
8. List the common logical operators.
9. List the three kinds of input statements offered by BASIC.
10. Give the syntaxes of the above statements.
11. Outline the program control statements we have in BASIC.
12. Give the syntaxes of the control statements.
13. Explain what you understand by library functions.
14. Give some examples of library functions.
15. What are the advantages of library functions to programmers?

Reading List

1. Schneider,¹ D. I. (1992). *QBASIC with an introduction to Visual Basic 5.0*. New Jersey: Pearson Hall, Upper Saddle River. ISBN 0-13-973876-2
2. Gayle³, A. (1992). *Building with BASIC: A programming kit for kids*. London: Alpha books. ISBN 0-672-30057-5.
3. BASIC Programming Tutorial³. Available at http://home.cmit.net/rwolbeck/programming_tutorial. Accessed 26th September, 2011.
4. The BASIC Programming Language³. Available at <http://groups.engin.umd.umich.edu/CIS/Course.des/cis400/basic/basic.html>. Accessed 26th September, 2011.
5. Lab: Computer Programming with QBASIC³. Available at <http://www.is.wayne.edu/gst2710/qbasic/qbasic.htm>. Accessed 26th September, 2011.
6. Basic Programming – QBasic Tutorial One. Available at <http://www.youtube.com/watch?v=etGDw2EccqI>. Accessed 26th September, 2011.

Week 15: Revision

Objectives

The main objective of the revision class is to provide explanation to the students on any topic or concept that they found difficult during the course of the lectures and practicals.

Description

First Hour

General review for examinations

Second Hour

Questions and Answers

Revision Questions

1. Discuss the role of the operating system in the computer.
2. Differentiate between the network operating system and the multi-user operating system.
3. Explain the following terms: compiler, source code, machine code.
4. Is it possible to detect logic errors during program compilation? Give reason for your answer.
5. Discuss any three characteristics of an algorithm
6. Demonstrate your understanding of algorithms and flowcharts by designing a flowchart to find the largest of any three numbers.
7. Compare and contrast algorithm and flowchart.
8. Define BASIC language along with its features.
9. Describe briefly the structure of BASIC language.
10. Differentiate between
 - (i) Numeric and String constant
 - (ii) Arithmetic and Relational expression
11. Write down the hierarchy of operations in arithmetic expression.
12. What is the difference between an IF... THEN statement and an IF....THEN...ELSE statement?
13. What will be the output of the following program?

```
10 FOR I = 1 TO 3
20 FOR J = 1 TO 5
30 PRINT I, J, I+J
40 IF J = 3 THEN 100
50 PRINT I, J, I*J
60 PRINT I + J, I-J
70 PRINT
100 NEXT J
110 NEXT I
120 END
```
14. Evaluate the following logical expressions:
 - (i) NOT (X AND Y) OR Z
 - (ii) (X AND Y) OR (Y AND Z)
15. Write a BASIC program to compute the area of a rectangle with the formula $A = L \times B$

General Reading List

1. French¹, ², C.S. (2001). *Introduction to Computer Science*. (5th Edn.). London: BookPower, 506p. ISBN 0-8264-6761-X.
2. Cormen³, T. H., Leiserson, C.E., Rivest, R. L. & Stein, C. (2001). *Introduction to algorithms*. USA: MIT Press, 1145p. ISBN 0-07-013151-1.
3. Tanenbaum³, A.S. (2001). *Modern Operating Systems*. (2nd Edn.). 935p.
4. Operating system – Wikipedia free encyclopedia³. Available at: <http://en.wikipedia.org/wiki/operating> system. Accessed 19th September, 2011.
5. What is an operating system? – operatingSystems.com³. Available at

- <http://www.operatingsystems.com/operating> systems. Accessed 19th September, 2011.
6. How operating systems works – HowStuffWorks.com³. Available at <http://www.HowStuffworks.com/operating> systems. Accessed 19th September, 2011.
 7. Compiler – Wikipedia free encyclopedia³. Available at: <http://en.wikipedia.org/wiki/compiler>. Accessed 19th September, 2011.
 8. Knuth^{1,3}, D.E. (1998). *The Art of Computer Programming; Semi Numerical Algorithms*. California: Addison-Wesley. ISBN 0-201-03822-6
 9. Insertion Sort – Wikipedia free encyclopedia³. Available at: http://en.wikipedia.org/wiki/Insertion_Sort. Accessed 19th September, 2011
 10. BASIC–Wikipedia free encyclopedia³. Available at: <http://en.wikipedia.org/wiki/BASIC>. Accessed 19th September, 2011.
 11. Schneider,¹ D. I. (1992). *QBASIC with an introduction to Visual Basic 5.0*. New Jersey: Pearson Hall, Upper Saddle River. ISBN 0-13-973876-2
 12. Gayle³, A. (1992). *Building with BASIC: A programming kit for kids*. London: Alpha books. ISBN 0-672-30057-5.
 13. BASIC Programming Tutorial³. Available at <http://home.cmit.net/rwolbeck/programming> tutorial. Accessed 26th September, 2011.
 14. The BASIC Programming Language³. Available at <http://groups.engin.umd.umich.edu/CIS/Course.des/cis400/basic/basic.html>. Accessed 26th September, 2011.
 15. Lab: Computer Programming with QBASIC³. Available at <http://www.is.wayne.edu/gst2710/qbasic/qbasic.htm>. Accessed 26th September, 2011.
 16. Basic Programming – QBasic Tutorial One. Available at <http://www.youtube.com/watch?v=etGDw2EccqI>. Accessed 26th September, 2011.

Legend

- 1- Available in the University Library
- 2- Available in local bookshops
- 3- Available on the Web
- 4- Personal collection
- 5- Departmental Library