LONDON SOUTH BANK UNIVERSITY

unit guide

Engineering Software 1

ECS-1-805

Department of Electrical, Computer
and Communications Engineering

Faculty of Engineering, Science
and the Built Environment

2004 / 2005 (Semester 2)

Stavros Dimitriou

become what you want to be

Unit Title: Engineering Software 1

Reference Number	ECS-1-805
Level	1
Credits	1
Study hours	150 hours: 12 hrs lecture/tutorial, 24 hrs workshop/assignment, 114 hrs independent study.
Pre-requisites	None
Faculty	Engineering, Science and the Built Environment
Department	Electrical, Computer and Communications Engineering
Unit Coordinator	Mr Stavros Dimitriou Room T820, Tel.: 0207 815 7580, email: dimitrsa@lsbu.ac.uk
Academic year	2003 - 2004

Aims and Objectives

Software systems are now ubiquitous. The main aim of the unit is to introduce you to software engineering with Java programming. The unit has several important objectives and you should make sure that you understand them; otherwise you won't be able to develop a "feel" for how you are getting on. One is for you to begin to appreciate the role and importance of software and computers in Electrical and Electronic Engineering. This should provide you with a good impetus to quickly become competent in their use. Another is for you to learn how to design and realise Java programs in a systematic engineering fashion, mainly in an engineering context. Another is for you to get confidence in your own ability to learn a computer language (and other computer skills) on your own, from books and other sources. As a modern engineer you will have to get to grips with a lot of different computer languages and other "tools" (e.g. UML graphical notations), from books and manuals of varying quality, and usually on your own. Perhaps most importantly the unit will be one of your first chances to learn how to do things like an engineer, and get used to using well tried engineering principles which apply to all engineering, and not just software.

Learning Outcomes

These are the things you ought to be able to do after (successfully) finishing the unit. Without listing things at the very lowest level of detail, these include:

- Ability to design modular programs of moderate size, from given specifications, and document them using the Department's own in-house standard.

- Ability to translate informal or SDS3 designs into Java, using all the Java control structures, all its simple data structures, arrays and records, but excluding the use of dynamic storage and more complex data structures.
- Use IBM PC (clone) computers under MS DOS and Windows, for all aspects of file handling, and running Java as an application.

Syllabus

The following is an extract from the Course Document for the BEng in Electrical and Electronic Engineering. It gives you an idea of the topics you will study in this unit.

High-level language programming:
Constant, type, variable, function declaration. Scope of variables and functions. Sequence, selection and iteration. Machine available types, integer, character, real. Ordinal and non-ordinal types. Structures, arrays and user-constructed objects.

Design Techniques:
Top-down, or functional decomposition (FD) techniques. Bottom-up or object oriented approaches. Use of diagrams as an abstract design language. A brief introduction to UML Model diagrams.

Documentation:
The use of the ESBE faculty standards of documentation for the specification, test and design of software.

Unit Structure

The unit consists of 7 topics:

1. Introduction
2. Data types, Operations and Control Structures in Java
3. Java Programming Algorithms
4. Functions, Methods and Arrays
5. Object Oriented Programming (OOP)
6. Software Documentation, Development and Management
7. Introduction to Graphical User Interface (GUI) programming

Unit Calendar

Study Area	Week No
Introduction	1
Data types, Operations and Control Structures in Java	2-3
Java Functions and Methods, Arrays and Tables	4-5
Java Programming Algorithms	6
Review and tutorial	7
Mini-Test	8
Object Oriented Programming (OOP)	9-10
Software Development and GUI Programming	11-12
Revision	13
Examination	14-15

Expansion of study areas:

1. Introduction

Introduction to unit, Unit Guide, mode of working. Introduction to what software for engineering is all about. Introduction to the basic of Java Programming.

Learning outcome

On completion of this section you will be able: To understand how the unit is going to run, how to use the study guide, and so to help you plan your weekly time, which is essential if you are to keep up with your work, and succeed; To be able to list and discuss most of the engineering uses of computers, and begin to see why Software is so important in engineering, and an engineering approach so important in Software; To understand at a very basic level how a computer works, in particular how it is driven by Software; To load, run, and save simple pre-written programs in the laboratory.

2. Data types, Operations and Control Structures in Java

Role of Software in engineering, simple ideas about the software lifecycle, a very informal design method, algorithms and Functional Decomposition. Java data types and operations. Java Control Structures – selections and iterations.

Learning outcome

On completion of this section you will have the ability: To use an informal design method for software, forming the basis for more formal standards (SDS3) later; to wrire Java programs to perform simple calculations, to understand the structure of a program, to know Java primitive data types and to use Java operators, to understand what control structures are, and read a simple Functional Decomposition (FD) or UML Diagram, to use loop statements to control the repetition of statements.

3. Java Programming Algorithm

Review of simple program structure. Introduction to Java programming algorithms.

Learning outcome

On completion of this section you will have the ability: to produce simple correct FD's or UML model diagrams for algorithms you understand; to design, and implement your own algorithm to solve certain problems.

4. Functions, Methods and Arrays

Introduction to Java functions and methods; creating and calling a method, passing parameters, the scope of local variables. Introduction to Java arrays and tables; declaring array variables, creating, initializing and processing arrays, tables, sorting and searching.

Learning outcome

On completion of this section you will: (a) understand the basic engineering principles; know how to design and implements a function or a method in Java program; be able to specify, design and implement your own Java program, (b) understand the concept of arrays, to learn how to pass arrays to methods, to become familiar with search and sorting algorithms.

5. Object Oriented Programming (OOP)

Introduction to OOP. Concepts of Objects and Classes; Object references, strings, class inheritance and interfaces.

Learning outcome

On completion of this section you will be expected to: (a) understand objects and classes, to understand object references, to use objects as array elements, to use UML graphical notations to describe classes and objects, (b) to learn how to process strings using the String class, (c) to understand the concept of class inheritance, to use interfaces to model weak inheritance relationships.

6. Software Documentation, Development and Management

Introduction to SDS3 ESBE documentation standard. To begin the development of critical analysis skills, in order to learn best from existing examples. Revision of course material by use of case study.

Learning outcome

On completion of this section you will have the ability: to learn the basics of using the SDS3 method; to identify those aspects of a software project, which in the real world would have commercial implications; to enable you to translate given SDS3 designs, FD's or UML's quickly into code; to gain experience of managing a project, and of framing problems so as to make best use of outside advice; to consolidate design and problem solving skills.

7. Software Development and GUI Programming

Program development in software engineering; introduction to GUI programming; creating graphics and user interfaces, applets.

Learning outcome

On completion of this section you will have the ability: (a) to become familiar with the process of program development in software engineering, to become familiar with GUI programming, to be able to describe the Java GUI API hierarchy, (b) to know various user interface components, to learn how to display images and (c) to become familiar with applets, to understand how the web browser controls and executes applets

Tutorial examples

Tutorial examples sheets will be handed out at the end of formal teaching of each study area.

Teaching and Learning Methods

Teaching is by 12 hours lectures/tutorials, and 24 hours of laboratory work. Lectures will cover all the main aspects of the subject matter in the unit. Printed material, which will include some lecture material and tutorial examples, will be provided. The laboratory exercises are designed to supplement the lectures. Lectures and laboratory experiments are treated as a unified body of work. In addition, you are required to carry out 114 hours of self study.

Assessment

There will be one 2-hour written examination (50%), one mini-test around week 8 (10%) and one assignment (20%). Additionally, each student is expected to maintain a log book on all the lab works and submit it at the end of teaching week 12 (20%). The log books will be examined periodically during the lab sessions. In addition to the log book, you will be required to do one assignment. You will be required to submit the assignment (will be specified in the early part of the semester) by the final submission date, which will be notified during the semester allowing you sufficient time to complete your work. You MUST submit your assignments, following the standard faculty procedure, to J200 between 10:00am and 4:00pm. Late submission will be penalized in accordance with the University regulation. The examination will be set on all lectured material, the coursework and directed reading/research done in the course of the unit.

Examination	50%
Mini test	10%
Coursework	40% (assignment: 20 marks, log-book: 20 marks)

Indicative Book List

1. Y. Daniel Liang, Introduction to Java Programming, 4th edition (2003), Prentice Hall
2. Judith Bishop, Java Gently, 3rd edition (2001), Addison-Wesley
3. Ian Sommerville; Software Engineering, 6th edition, Addison Wesley
4. Mary Campione, Kathy Walrath, The Java Tutorial Second Edition, Addison-Wesley, 1999
5. D.Bell, I.Morrey, J.Pugh, Software Engineering, A Programming Approach, Prentice Hall International, 1987.
6. Stephen J. Chapman, Java for engineers and scientists, Prentice Hall, 2000

Background Reading

1. H.M. Deitel, P.J. Deitel, Java How to Program, Prentice-Hall, 1999
2. Laura Lemay, Rogers Cadenhend, Sams Teach Yourself Java 2 in 21 Days, Sams, 1999
3. Walter Savitch, Java - A Introduction to Computer Science & Programming, Prentice-Hall 1999
4. Fintan. Culwin, Java : an object first approach, Prentice-Hall 1998
5. Java Online Tutorials: http://java.sun.com/docs/books/tutorial/index.html
6. Java Platform Documentation: http://java.sun.com/docs/
7. Java Programming: http://www.meurrens.org/ip-Links/java/codeEngineering/
8. Java sources: http://www.developer.com/

Study Hours

You may notice that this guide states that the unit requires 150 study hours, whereas previous guides have defined each unit as 120 study hours. The University has made this change in line with the way study time is likely to be expressed, in future, in the majority of Universities. There is no change in teaching time and no change in what you are expected to do or achieve. The change concerns the way study time is measured. Previously, the unit was defined as 120 hours work over 12 teaching weeks. The new measure is still 10 hours per week over 15 weeks, including assessment.

The workload for a full time student is still expected to be approximately 40 hours per week.