Philadelphia University
Faculty of Information Technology
Department of Software Engineering

	Course Syllabus
Course Title: Developing Computer Aided Software Engineering	**Course code: 0721445**
Course Level: 4th year of study	**Course prerequisite (s): 07213420**
Lecture Time:	**Credit hours: 3**

		Academic Staff Specifics		
Name	**Rank**	**Office Number and Location**	**Office Hours**	**E-mail Address**

Course module objectives:

The aim of the module is to develop understanding of and gain experience with the basic techniques of software engineering with the help of computer-aided software. The course provide an introduction of life cycle models, various case work benches using in an integrated environments followed by hands on experience on CASE TOOLS. Also, techniques of developing CASE tools that match the needs of software engineers will be addressed.

Course/ module components

- Stephen R.Schach,Classical and Object Oriented Software Engineering, 4th edition, McGRAW-Hill, USA 1999
- Ian Sommerville, Software Engineering, seventh edition, Addison –Wesley, 1996.
- Jeffrey Hoffer,Joey George, Joseph Valacich, Modern System Analysis & Design ,3rd edition ,2002.

The course intends to cover the following main areas of CASE Tools:

- **Software Engineering Overview, Software process and Life cycle**
- **CASE Technology ,CASE Classifications**
- **Workbenches , Integrated Environment for Case Tools**
- **Project Management Support CASE Tools**
- **CASE Tools for project planning and Estimation**
- **CASE Support Requirement Engineering**
- **CASE Support Design**
- **Coding Support CASE Tools:**
- **CASE Support Testing:**
- **Configuration Management CASE Tools**
- **Programming & Building a CASE Too**

Teaching methods:

Duration: 16 weeks in the semester, 48 hours in total
Lectures and Tutorials: 36 hours, 3 per week (plus 1-2 hours midterm exam).
Project: 09 hours class works

Learning outcomes:

A student completing this module unit should be able to:

- **Knowledge and understanding**

1. Recognize and understand a wide range of principles and CASE tools available to the software engineer and information system developer in particular, such as data bases, Analysis/design/coding/testing/maintaining methodologies, and user interface techniques. (A)
2. Recognize and understand the professional and ethical responsibilities of the practicing CASE tools users. (A)
3. Recognize and understand the application of CASE tools in software development (A)

- **Cognitive skills (thinking and analysis).**

1. Solve a wide range of problems related to the analysis, design and construction of information systems using CASE tools. (B)
2. Survey, design, and implement CASE tools of different purposes. (B)
3. Plan and undertake a major individual project, and prepare and deliver coherent and structured written technical report with the aid of CASE tools. (C)

- **Communication skills (personal and academic)**

1. Prepare and deliver coherent and structured verbal and written technical report..
2. Display an integrated approach to the deployment of communication skills, use IT skills and display mature computer literacy; strike the balance between self-reliance and seeking help when necessary in new situations, and display personal responsibility by working to multiple deadlines in complex activities

- **Practical and subject specific skills (Transferable Skills).**

1. Display an integrated approach to the deployment of communication skills, use IT skills and display mature computer literacy, strike the balance between self-reliance and seeking help when necessary in new situations, and display personal responsibility by working to multiple deadlines in complex activities. (C)

Assessment instruments

Allocation of Marks	
Assessment Instruments	**Mark**
First examination	**20**
Second examination	**20**
Final examination:	**40**
Reports, research projects, Quizzes, Home works, Projects	**20**
Total	**100**

Documentation and academic honesty

- Documentation style: University rules apply.
- Protection by copyright

1. You must not make an unauthorized copy, in any form, of copyright software or data.
2. Note that this does not prevent your taking copies of your laboratory work home, or making copies of non-copyright material, but does prevent your taking random pieces of software away on a floppy. You should assume that all material is copyright unless it specifically states otherwise. If in doubt, ask.

- Avoiding plagiarism.

1. Coursework, laboratory exercises reports and essays submitted for assessment must be your own work, unless in the case of group projects a joint effort is expected and is indicated as such.
2. Unacknowledged direct copying from the work of another person, or the close paraphrasing of somebody else's work, is called plagiarism and is a serious offence, equated with cheating in examinations. This applies to copying both from other students' work and from published sources such as books, reports or journal articles.
3. Use of quotations or data from the work of others is entirely acceptable, and is often very valuable provided that the source of the quotation or data is given. Failure to provide a source or put quotation marks around material that is taken from elsewhere gives the appearance that the comments are ostensibly your own. When quoting word-for-word from the work of another person quotation marks or indenting (setting the quotation in from the margin) must be used and the source of the quoted material must be acknowledged.
4. Paraphrasing, when the original statement is still identifiable and has no acknowledgement, is plagiarism. A close paraphrase of another person's work must have an acknowledgement to the source. It is not acceptable for you to put together unacknowledged passages from the same or from different sources linking these together with a few words or sentences of your own and changing a few words from the original text: this is regarded as over-dependence on other sources, which is a form of plagiarism.
5. Direct quotations from an earlier piece of your own work, if not attributed, suggest that your work is original, when in fact it is not. The direct copying of one's own writings qualifies as plagiarism if the fact that the work has been or is to be presented elsewhere is not acknowledged.
6. Sources of quotations used should be listed in full in a bibliography at the end of your piece of work.
7. Plagiarism is a serious offence and will always result in imposition of a penalty. In deciding upon the penalty the Department will take into account factors such as the year of study, the extend and proportion of the work that has been plagiarized and the apparent intent of the student. The penalties that can be imposed range from a minimum of a zero mark for the work (without allowing resubmission) through caution to disciplinary measures (such as suspension or expulsion).

Course/module academic calendar

Week	Subject	
1.	**Software Engineering Overview, Software process and Life cycle**	
2.	**CASE Technology ,CASE Classifications**	
3.	**Workbenches , Integrated Environment for Case Tools**	
4.	**Project Management Support CASE Tools**	
5.	**1)CASE Tools for project planning and Estimation** **2)Lab& Tutorial :** Project Management with Ms Project 2002	
6.	**CASE Support Requirement Engineering** **FIRST EXAM**	
7.	1) Requirement Work flow 2) Requirement management ,Case Tools Support for Requirement Management **3) Lab& Tutorial :** Requirement Management case Tools Rational RequisitePro	
8.	**CASE Support Design** 1) Introduction to Object Oriented design 2) UML as a popular OODesign Tool	
9.	**Coding Support CASE Tools:** 1) Code Generating Case Tools **2) Lab& Tutorial :** Analysis - Design – Implementation (Rational Rose, Ms Visio)	
10.	**CASE Support Testing:** Factors controlling CASE tools development	

11.	Customized SDLC for CASE tools development **SECOND EXAM**	
12.	Programming languages and CASE tools development	
13.	Software Testing and Testing Techniques **Lab& Tutorial :** Rational Test*	
14.	**Configuration Management CASE Tools**	
15.	**Programming & Building a CASE Tool**	
16.	**Case Study**: Building a Requirement CASE Tools **FINAL EXAM**	

Expected workload:

On average students need to spend 2 hours of study and preparation for each 50-minute lecture/tutorial.

Attendance policy:

Absence from lectures and/or tutorials shall not exceed 15%. Students who exceed the 15% limit without a medical or emergency excuse acceptable to and approved by the Dean of the relevant college/faculty shall not be allowed to take the final examination and shall receive a mark of zero for the course. If the excuse is approved by the Dean, the student shall be considered to have withdrawn from the course.

Module references

1. Stephen R.Schach,Classical and Object Oriented Software Engineering, 4th edition, McGRAW-Hill, USA 1999
2. Ian Sommerville, Software Engineering, seventh edition, Addison –Wesley, 1996.
3. Jeffrey Hoffer,Joey George, Joseph Valacich, Modern System Analysis & Design ,3rd edition ,2002.

Websites

1. http://osiris.sunderland.ac.uk/sst/casehome.html
2. http://www.hotbot.lycos.com/?comefrom=nspanel-search&MT=case+tools
3. http://www.qucis.queensu.ca/Software-Engineering/tools.html