QFO-AP-FI-MO02	**اسم النموذج: Course Syllabus**	**جامعة فيلادلفيا** **Philadelphia University**
رقم الاصدار : (Revision) 1	**الجهة المصدرة: كلية تكنولوجيا المعلومات**	
التاريخ :2017/11/05	**الجهة المدققة: عمادة التطوير والجودة**	
عدد صفحات النموذج:		

Course Title: Concepts of Programming Languages	**Course Code:** 0750321
Course Level: 3	**Course Prerequisite:** 0721224
Lecture Time:	**Credit Hours:** 3

Academic Staff Specifics

Name	Rank	Office Number and Location	Office Hours	e-mail Address

Course Description:

This course is based on the textbook Concepts of Programming Languages, Tenth Edition. It aims to provide the student with a framework for thinking about programming languages. It introduces different programming paradigms through many concepts of different programming languages.

Course Objectives:

There are always new languages being devised, of which a very few actually become widely used; typically, these are specialized languages for particular applications (e.g., Java). As a computer scientist, the student must be able to learn new languages as necessary, and the background he/she gets from this course should make this easier. Finally, students will almost certainly have to choose which programming language to use for a particular project. A final goal of this module is to give students enough background in the study of programming languages that they can argue persuasively why a particular language is appropriate (or inappropriate) for a particular problem.

Course Components:

Generalities on programming languages and paradigms

Names, Bindings, and Scopes in various programming languages and paradigms

Data Types, type checking, primitive Data Types, user-defined Data Types, Character String Types, Array Types, Record Types, Union Types, Set Types, Pointer Types, Theory and Data Types

Expressions and Assignment Statements

Statement-Level Control Structures

Subprograms and their implementation Abstract Data Types

Functional Programming

Logic Programming

Scripting Languages

Textbook and Supporting Material:

Concepts of Programming Languages, Tenth Edition, Robert W. Sebesta, Pearson, 2013
the students will be provided with handouts by the lecturer.

Teaching Methods:

Lectures, tutorial, lab work, discussion groups.

Learning Outcomes:

A. Knowledge and Understanding

A1. Understanding different programming paradigms
A2. Understanding the syntax and semantic of programming languages

B. Intellectual Skills

B1. Comparing between various languages based on the same paradigm
B2. Comparing between various languages based on different paradigms B3. Designing a new language

C. Practical skills

C1. Developing different projects using different programming languages
C2. Applying the “right” language to the “right” application and/or project

D. Transferable Skills

D1. Preparing structured technical reports for the assigned lab works
D2. Delivering verbal communication on the performed assignments

Learning outcomes achievement:

Development

A1, A2, B1, B2 are developed through lectures and assignments
B3, C1, C2 are developed through assignments and lab work
D1 and D2 are assessed by tutorials and assignments

Assessment

A1, A2, B1, B2 are assessed through quizzes and examinations
B3, C1, C2, D1 and D2 are assessed through assignments and lab work

Assessment Instruments:

Allocation of Marks	
Assessment Instruments	**Mark**
First exam	20%
Second exam	20%
Final Exam	40%
Reports, Assignments, Quizzes, Discussion	20%
Total	100%

** Make-up exams will be offered for valid reasons only with consent of the Dean. Make-up exams may be different from regular exams in content and format.*

Practical Submissions
The assignments that have work to be assessed will be given to the students in separate documents including the due date and appropriate reading material.

Documentation and Academic Honesty
Submit your homework covered with a sheet containing your name, number, course title and number, and type and number of the home work (e.g. tutorial, assignment, and project).

Any completed homework must be handed in to my office (room 601 IT building) by 15:00 on the due date. After the deadline a nil mark will be awarded. You must keep a duplicate copy of your work because it may be needed while the original is being marked.

You should hand in with your assignments:

1- A printed listing of your test programs (if any).
2- A brief report to explain your findings.
3- Your solution of questions.

For the research report, you are required to write a report similar to a research paper. It should include the following:

- **An abstract**: It describes the main synopsis of your paper.
- **An introduction**: It provides background information necessary to understand the research and getting readers interested in your subject. The introduction is where you put your problem in context and is likely where the bulk of your sources will appear.
- **A description of the methods (Algorithms and Implementation)**: Describe your methods here. Summarize the algorithms generally, highlight features relevant to your project, and refer readers to your references for further details.
- **The results and a discussion (Benchmarking and Analysis)**: This section is the most important part of your paper. It is here that you demonstrate the work you have accomplished on this project and explain its significance. The quality of your analysis will impact your final grade more than any other component on the paper. You should therefore plan to spend the bulk of your project time not just gathering data, but determining what it ultimately means and deciding how best to showcase these findings.
- **A conclusion**: The conclusion should give your reader the points to “take home” from your paper. It should state clearly what your results demonstrate about the problem you were tackling in the paper. It should also generalize your findings, putting them into a useful context that can be built upon. All generalizations should be supported by your data, however;

the discussion should prove these points, so that when the reader gets to the conclusion, the statements are logical and seem self-evident.

- **The bibliography:** Refer to any reference that you used in your assignment. Citations in the body of the paper should refer to a bibliography at the end of the paper.

Protection by Copyright

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. 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.
3. Sources of quotations used should be listed in full in a bibliography at the end of your piece of work.

Avoiding Plagiarism.

1. 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.
2. 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.
3. 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.
4. 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 extent 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 Academic Calendar

Week	**Basic and support material to be covered**	**Assignments / reports and their due dates**
(1)	Programming Domains, Language Evaluation Criteria, Influences on Language Design, Language Categories, Language Design Trade-Offs, Implementation Methods, Programming Environments	
(2)	Names, Binding and Scopes: Names, Variables, The Concept of Binding	
(3)	Names, Binding and Scopes: Scope and Lifetime, Referencing Environments, Named Constants	
(4)	Data Types: Primitive Data Types, Character String Types, User-Defined Ordinal Types	Assignment 1
(5)	Data Types: Array Types, Record Types, List Types, Union Types	Tutorial 1
(6) **First Exam**	Data Types: Pointer and Reference, Type Checking, Strong Typing, Type Equivalence, Theory and Data Types.	
(7)	Expressions and Assignment Statement: Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions	Assignment 1 due date Assignment 2
(8)	Expressions and Assignment Statement: Short-Circuit Evaluation, Assignment Statements, Mixed-Mode Assignment	
(9)	Statement-Level Control Structures: Selection Statements, Iterative Statements	
(10)	Statement-Level Control Structures: Unconditional Branching, Guarded Commands	Assignment 2 due date Assignment 3
(11)	Subprograms: Fundamentals of Subprograms, DesignIssuesforSubprograms,Local Referencing Environments	Tutorial 2
(12) **Second Exam**	Subprograms: Parameter-Passing Methods, Parameters That Are Subprograms, Calling Subprograms Indirectly	
(13)	Subprograms: Overloaded Subprograms, Calling Subprograms Indirectly, Generic Subprograms,	Assignment 3 due date
(14)	Subprograms: Design Issues for Functions, User-Defined Overloaded Operators, Closures, Coroutines	
(15)	Logic Programming, Functional Programming	Tutorial 3
(16) **Final Examination**	Scripting Languages	

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.

Other References:

1- Terrence W. Pratt, Programming Languages: Design and Implementation, Prentice-Hall, 2002
2- Ravi Sethi, Programming Languages Concepts and Constructions, Pearson Education, 1996 3- Allen B. Tucker, Programming Languages, McGraw Hill, 1988
4- C. Ghezzi and M. Jazayeri, Programming Language Concepts, John Wiley and Sons, 1982

WEB LINKS

1. http://its.metu.edu.tr/
2. http://courses.cs.washington.edu/courses/cse505/03au/lec1.pdf
3. http://www.cs.bu.edu/~hwxi/academic/courses/CS320/Fall04/notes/lectures.pdf
4. http://www.cs.nccu.edu/~melikyan/cs_prl/lect/lec/TextNotes.html
5. http://homes.ieu.edu.tr/bhnich/CS305/index.html
6. http://www2.cs.uidaho.edu/~jeffery/courses/210/lecture.html
7. http://us.cambridge.org/features/0521780985/
8. http://www.cs.rochester.edu/u/scott/pragmatics/
9. http://cs.fit.edu/~ryan/cse4250/
10. http://www.cmi.ac.in/~madhavan/courses/pl2005/lecturenotes/lecture-notes.pdf