Information Technology - intermediate - syllabus

Course period: 4th module (April - May 2018)
Course description: This intermediate course is designed to give practical knowledge, and considers mostly “the computer as a tool”. The course is designed assuming that that the student has already some working knowledge of computers and the internet.
It will still start with some brief theoretical general definitions about computers hardware and software.
It will then focus on some advanced use of software tools for: Word processing, Spreadsheet, Presentation, Scientific Typesetting, Graphic editing.
It will then give some theoretical notions for sound code design, illustrating the different coding approaches. Following, some extensive knowledge on programming, with focus on applications and concrete examples. The programming will be done mostly in Python language, with some examples of C language. The programming will be done in practical sessions in the computer room.
The aim of this course: To give to the students a theoretical base of concepts to correctly understand what is a computer, how it works, and what are its potentials and its limits. To give the knowledge, the skills, and enough hands-on experience, to be able to write good computer code, for a large spectrum of applications, with a particular focus on Social Sciences and Humanities.
Objectives:

- Basic computer hardware understanding and troubleshooting.
- Basic understanding of concepts and technical definitions of Computer science and computer networks.
- Discrete knowledge of programming techniques, with emphasis on applications on Social Sciences and Humanities.
- Python language programming, theory and practice.

The course structure (number of lectures, seminars): The course will develop over 8 weeks, with 32 classes of 90’, of which **12** will be in the form of lectures (frontal teaching) and **20** will be in the form of Computer Lab Practice. There will be ? lectures and ? seminars per week. Most of the seminars will be conducted in a computer room, where each student will be seating in front of a computer.
Author of the syllabus: Fabio Grazioso
Lecturer (name and surname): Fabio Grazioso
Seminar’s professor (name and surname): Fabio Grazioso
Course language: English
Course requirements: Basic knowledge of computer usage, MS-Windows operations. No advanced computer knowledge and/or experience will be assumed. Mathematics (algebra) knowledge is required.
Ground rules: Attendance to 85% of the lectures is required. 1 intermediate assignment (homework) and 1 final assignment will be required.
Sample exam’s questions: What is a general definition of an Operative System? What is the meaning of “data packet” talking of computer network protocols? What is the name of the graphic library used for data plotting in Python? What is a regular expression? What is the difference between direct methods and iterative methods in Numerical Analysis?
Reference Textbooks and other material:

1) Fox - Information Technology: An Introduction for Today’s Digital World (2013).
2) Vermaat, Sebok, Freund - Discovering Computers 2017 (2016).
3) Tanenbaum, Bos - Modern Operating Systems (2014).
4) Harris, Harris - Digital Design and Computer Architecture (2nd Ed) (2012).
5) Mueller - Windows command line administration instant reference (2010).
6) Gillam - Unicode Demystified (2002).

7) Hunt - TCP-IP network administration (2002).
8) Martin - Clean Code- A Handbook of Agile Software Craftsmanship (2008).
9) Cormen - Algorithms unlocked (2013).
10) Sauer - Numerical Analysis (2011).
11) Atkinson - An Introduction to Numerical Analysis (1989).
12) Burden, Faires, Burden - Numerical Analysis (2016).
13) Papoulis, Pillai - Probability, Random Variables and Stochastic Processes (2002).
14) Jaynes - Probability Theory The Logic of Science (2003).
15) Lutz - Learning Python (2013).
16) Langtangen - A primer on scientific programming with Python (2012).
17) Friedl - Mastering Regular Expressions (2006).
18) Baecker - Readings in human-computer interaction (1995).
19) Norman - The Design of Everyday Things (2002).

(note: the books list is very long, but from most of the books only small parts, sometimes just few pages, will be considered.)

In the following you can find a summary table of the main topics and the detailed schedule.

Information Technology intermediate - course summary

General hardware and software definitions. General architecture of a computer. The machine representation of numbers and errors. The operating system. MS-DOS, Microsoft Windows, Apple MacOS.
Application software and Utility software. Word processors. Spreadsheets. Presentation. Scientific Typesetting, Graphic editors. Audio editors. Text editors. Utility software, examples.
Networks fundamentals. World Wide Web, HTML. Search engines. Other protocols: email, FTP, servers, clients.
Introduction to programming. Software design techniques. Algorithms. Methods for code development: top-down and bottom-up. Main programming statements and pseudo-code. Procedural programming, structured programming and object-oriented programming.
Programming languages. Main programming language types: machine code, low-level, high-level, interpreted, compiled. Syntax and semantics.
Elements of the C language. Keywords and syntax. Control flow statements. Data types. Pointers. Code examples.
The Python language. Keywords and syntax. Control flow statements. Data types. Libraries: NumPy, SciPy, Matplotlib. Examples of data processing. Examples of data plotting and representation: 2D plots, 3D plots, annotations. Datafiles input and output. Strings manipulation. Regular Expressions.
Numerical analysis. Direct methods and iterative methods. Precision. Example algorithms: sorting, searching, root-finding, linear equation systems solving, functions interpolation, functions integration.

Statistics and Probability.
Fundamentals of statistics. Fundamentals of probability theory. Bayes theorem.

Curse classes

(type: L= lecture, P = practice)

#	type	topics
1	L	General architecture of a computer. The Von Neumann machine. The machine representation of numbers and errors (logical). The binary numbers, the hexadecimal numbers. Encodings. The operating system in general. MS-DOS, Graphical Operating Systems: MS Windows MacOS.
2	L	Word processors. Spreadsheets. Presentation. Graphic editors. Audio editors. Text editors. Utility software, examples.
3	P	Computer Lab practice on word processing, spreadsheets, presentation software.
4	P	Computer Lab practice on graphic editing (both vector and raster) for academic applications.
5	P	Computer Lab practice on audio editing and Scientific Typesetting (LaTeX)
6	L	Networking protocols, layers, packets. TCP/IP protocol. World Wide Web, HTML, browsers. Search engines. Other protocols: email, FTP, servers, clients.
7	P	Computer Lab practice on internet resources: academic resources, advanced search engines practice.
8	L	Main programming language types: machine code, low-level, high-level, interpreted, compiled. Syntax and semantics.
9	L	Programming environments: editor, debugger, compiler, interpreter. Software design techniques. Algorithms.
10	L	The Python language. Keywords and syntax. Control flow statements. Data types.
11	P	Computer Lab practice on Python programming: first simple algorithms.
12	L	Python Libraries: NumPy, SciPy. Examples of mathematical functions, mathematical data generation and manipulation. Examples of simple scientific plots.
13	P	Computer Lab practice on Python programming: first scientific applications.
14	P	Computer Lab practice on Python programming: more scientific applications.
15	L	Python data processing. Examples of data plotting and representation: 2D plots, 3D plots, annotations.

16	P	Computer Lab practice on Python programming: scientific data plotting.
17	P	Computer Lab practice on Python programming: scientific data plotting.
18	L	Python datafiles input and output. Text files and binary files. Strings Manipulation. Regular Expressions.
19	P	Computer Lab practice on Python programming: data files input and output.
20	L	Numerical analysis. Direct methods and iterative methods. Precision. Example algorithms: sorting, searching.
21	L	More numerical analysis. Root-finding, linear equation systems solving, functions interpolation, functions integration. Discrete Mathematics.
22	P	Computer Lab practice on Python programming: applications on Numerical analysis and discrete mathematics.
23	P	Computer Lab practice on Python programming: applications on Numerical analysis and discrete mathematics.
24	L	Statistics and Probability. Fundamentals of statistics. Fundamentals of probability theory. Bayes theorem.
25	P	Computer Lab practice on Python programming: applications on statistics.
26	P	Computer Lab practice on Python programming: applications on statistics.
27	P	Computer Lab practice on Python programming: Examples from the course.
28	P	Computer Lab practice on Python programming: Examples from the course.
29	P	Computer Lab practice on Python programming: Examples from the course.
30	P	Computer Lab practice on Python programming: Examples from the course.
31	P	Computer Lab practice on Python programming: Examples from the course.
32	P	Computer Lab practice on Python programming: Examples from the course.