

Course No.	Course Name	L-T-P-Credits	Year of Introduction
BE101-05	INTRODUCTION TO COMPUTING AND PROBLEM SOLVING	2-1-0-3	2015
Course Objectives			
1. To learn basics of digital computers 2. To develop problem solving skills 3. To learn programming and to solve problems using computers			
Syllabus Introduction to digital computer, Introduction to programming languages, Operating systems, Problem Solving strategies, Examples for algorithms and flow charts, Introduction to Python language, functions, parameters and arguments, Boolean Expressions, logical operators and control statements Strings, lists, tuples and dictionaries, operations, Files, introduction to objects, attributes and instances			
Expected outcome			
1. Ability to design algorithmic solution to problems. 2. Ability to convert algorithms to Python programs. 3. Ability to design modular Python programs using functions 4. Ability to design programs with Interactive Input and Output, utilizing arithmetic expression repetitions, decision making, arrays. 5. Ability to design programs using file Input and Output. 6. Ability to develop recursive solutions.			
Text Books:			
<ul style="list-style-type: none"> • Downey, A. et al., How to think like a Computer Scientist: Learning with Python, John Wiley, 2015 • Goel, A., Computer Fundamentals, Pearson Education • Lambert K. A., Fundamentals of Python - First Programs, Cengage Learning India, 2015 • Rajaraman, V., Computer Basics and C Programming, Prentice-Hall India 			
References Books:			
<ul style="list-style-type: none"> • Barry, P., Head First Python, , O' Reilly Publishers • Dromy, R. G., How to solve it by Computer, Pearson India • Guzdial, M. J., Introduction to Computing and Programming in Python, Pearson India • Perkovic, L., Introduction to Computing Using Python, 2/e, John Wiley, 2015 • Sprankle , M., Problem Solving & Programming Concepts, Pearson India • Venit, S. and Drake, E., Prelude to Programming: Concepts & Design, Pearson India • Zelle, J., Python Programming: An Introduction to Computer Science, Franklin, Beedle & Associates Inc. 			

Web links:			
Module	Contents	Hours	Sem. Exam Marks
I	<p>Introduction to digital computer – Von Neumann concept – A simple model of computer, acquisition of data, storage of data, processing of data, output of processed data. Details of functional units of a computer. Storage – primary storage and secondary storage.</p> <p><i>(The discussion should focus more on the functionalities of the units and their interaction than on specific hardware details. However, concepts like memory cells and their addressability (need not be binary), registers, interconnections (buses) have to be introduced at an abstract level. For storage devices – primary and secondary –, various categories have to be introduced along with their distinguishing features. For I-O devices also, various categories are to be introduced. The Von Neumann concept should be effectively introduced. History computers need not be taught. However, students have to be encouraged to read the relevant sections of the text book. Chapters 1 – 4 of ‘Goel’ may be used to support teaching -learning.)</i></p> <p>Introduction to programming languages:- types of programming languages - high level language , assembly language and machine language, System software - Operating systems – objectives of operating systems, compiler, assembler and interpreter.</p> <p><i>(For all the above topics, focus should be more on the concepts, significance and objectives. Chapter 6 and 7 (up to 7.4) of ‘Goel’ may be used to support the teaching-learning process.)</i></p>	8	15%
II	<p>Problem Solving strategies – Problem analysis – formal definition of problem – Solution – top- down design – breaking a problem into sub problems- overview of the solution to the sub problems by writing step by step procedure (algorithm) - representation of procedure by flowchart - Implementation of algorithms – use of procedures to achieve modularity.</p> <p><i>(For this part the instructor has to initially use suitable analogies of real world problems to explain the concepts, before delving into computer-solvable problems.)</i></p> <p>Examples for algorithms and flow charts - at least 10 problems (starting</p>	8	15%

	with non-numerical examples, and numeric problems like factorial, largest among three numbers, largest among N, Fibonacci etc.; <i>to be introduced with progressive levels of difficulty</i>) must be discussed in detail. (<i>Class assignments and/or tutorials may be used to strengthen understanding of this part. Chapters 4 and 5 of the 'Rajaraman' may be used for the teaching-learning process.</i>)		
--	---	--	--

FIRST INTERNAL EXAM

III	Introduction to Python – variables, expressions and statements, evaluation of expressions, precedence, string operations <i>(Note:- the instructor can demonstrate simple programs to the students and encourage them to develop similar ones. In particular, before attempting programs containing functions, the students should be given enough support and time to develop python code containing long sequence of statements for the simple flowcharts developed earlier. This will strengthen the students' understanding of instruction sequencing. Chapters 1 and 2 of 'Downey' have to be covered. Chapter 1 & 2 of 'Lambert' can also be used.)</i> Control statements, Boolean expressions and logical operators, conditional and alternative executions <i>(Note: - Chapter 4 of 'Downey' up to Section 4.9 has to be covered. The instructor should demonstrate each of these concepts with real examples and encourage students to develop as many as possible. Chapter 3 of 'Lambert' can be used for detailed discussion and self-study)</i> Iteration - while statement and tables. <i>(Note: - Chapter 6 of 'Downey' has to be covered. Chapter 3 of 'Lambert' can be used for detailed discussion and self-study.)</i>	8	15%
IV	Functions, calling functions, type conversion and coercion, composition of functions, mathematical functions, user-defined functions, parameters and arguments. <i>(Note: - Chapter 3 of 'Downey' has to be covered. The instructor should demonstrate each aspect of the function with real examples and encourage students to develop their own. Chapter 6 (up to 6.3) of 'Lambert' can be used for detailed discussion and self-study.)</i>	6	15%

SECOND INTERNAL EXAM

V	Strings and lists – string traversal and comparison with examples. <i>(Note: - Chapter 7 of 'Downey' has to be covered. Section 4.1 of 'Lambert' can be used for detailed discussion and self-study.)</i> List operations with examples <i>(Note: - Chapter 8 of 'Downey' up to Section 8.6 has to be covered. Section 5.1 of 'Lambert' can be used for detailed discussion and self-study.)</i> ; tuples and dictionaries – operations and examples <i>(Note: -</i>	6	20%
----------	---	---	-----

	<i>Chapters 9 & 10 of the third text have to be covered. Section 5.4 of 'Lambert' can be used for detailed discussion and self-study.)</i>		
VI	Files and exceptions - text files, directories <i>(Note: - Chapter 11 of 'Downey' has to be covered)</i> Introduction to classes and objects - attributes, instances <i>(Note: - Chapter 12 of 'Downey' up to Section 12.6 has to be covered)</i>	6	20%
END SEMESTER EXAM			