Syllabus for B.Sc.: Subject: Computer Science

Programme/Class: **Certificate**	Year: **First**	Semester: **First**
Subject: **Computer Science**		
Course Code: B070101T	Course Title: **Problem Solving using Computer**	

Course outcomes:

CO 1: Understand hardware components of computer system such as memory system organization, input/output devices, aware of software components of computer system, and windows operating system concepts.

CO 2: Develops basic understanding of computers, the concept of algorithm and algorithmic thinking.

CO3: Develops the ability to analyze a problem, develop an algorithm to solve it.

CO4: Develops the use of the Python programming language to implement various algorithms, and develops the basic concepts and terminology of programming in general.

CO5: Introduces the more advanced features of the Python language

Credits: **4**	**Core Compulsory**
Max. Marks: **25+75**	Min. Passing Marks:

Total No. of Lectures-Tutorials-Practical (in hours per week): **4-0-0**

Unit	Topic	No. of Lectures
I	**Computer Fundamentals:** Introduction to Computers: Characteristics of Computers, Uses of computers, Types and generations of Computers.	7
II	**Basic Computer Organization** - Units of a computer, CPU, ALU, memory hierarchy, registers, I/O devices. **Planning the Computer Program:** Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation.	8
III	**Techniques of Problem Solving:** Flowcharting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming.	7
IV	**Overview of Programming:** Structure of a Python Program, Elements of Python, IDEs for python, Python Interpreter, Using Python as calculator, Python shell, Indentation.	8
V	**Introduction to Python:** Atoms, Identifiers and keywords, Literals, Strings, Operators (Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment, Operator, Ternary operator, Bit wise operator, Increment or Decrement operator).	8
VI	**Creating Python Programs:** Input and Output Statements, Control statements (Looping- while Loop, for Loop, Loop Control, Conditional Statement- if...else, Difference between break, continue and pass).	7
VII	**Structures**: Numbers, Strings, Lists, Tuples, Dictionary, Date & Time, Modules, Defining Functions, Exit function, default arguments. File handling in python.	7
VIII	**Introduction to Advanced Python:** Objects and Classes, Inheritance, Regular Expressions, Event Driven Programming, GUI Programming. Basic concepts of concepts of Package and modules	8

Suggested Readings:

1. P. K. Sinha & Priti Sinha , "Computer Fundamentals", BPB Publications, 2007.
2. Dr. Anita Goel, Computer Fundamentals, Pearson Education, 2010.
3. T. Budd, Exploring Python, TMH, 1st Ed, 2011
4. Python Tutorial/Documentation www.python.or 2010
5. Allen Downey, Jeffrey Elkner, Chris Meyers , How to think like a computer scientist : learning with Python , Freely available online.2012
6. Rober Sedgewick, K Wayne -Introduction to Programming in Python: An interdisciplinary Approach" Pearson India

Suggestive digital platforms web links-

Programme/Class: **Certificate**	Year: **First**	Semester: **First**
Subject: Computer Science		
Course Code: B070102P	**Course Title: Software Lab using Python**	

Course outcomes:

1. To learn and understand Python programming basics.
2. To learn and understand python looping, control statements and string manipulations.
3. Students should be made familiar with the concepts of GUI controls and designing GUI applications.
4. To learn and know the concepts of file handling, exception handling and database connectivity.

Credits: **2**	Max. Marks: **25+75**	Min. Passing Marks:
Total No. of Lectures-Tutorials-Practical (in hours per week): **0-0-4**		

Suggested Readings:

1. Allen B. Downey, "Think Python: How to Think Like a Computer Scientist", 2nd edition, Updated for Python 3, Shroff/O'Reilly Publishers, 2016 (http://greenteapress.com/wp/thinkpython/)
2. Guido van Rossum and Fred L. Drake Jr, "An Introduction to Python – Revised and updated for Python 3.2, Network Theory Ltd., 2011.
3. Charles Dierbach, "Introduction to Computer Science using Python: A Computational Problem-Solving Focus, Wiley India Edition, 2013.
4. John V Guttag, "Introduction to Computation and Programming Using Python'', Revised and expanded Edition, MIT Press , 2013
5. Kenneth A. Lambert, "Fundamentals of Python: First Programs", CENGAGE Learning, 2012.

Section: A (Simple programs)

1. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon user's choice.
2. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria :
   Grade A: Percentage >=80
   Grade B: Percentage>=70 and <80
   Grade C: Percentage>=60 and <70
   Grade D: Percentage>=40 and <60
   Grade E: Percentage<40
3. Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user.
4. WAP to display the first n terms of Fibonacci series.
5. WAP to find factorial of the given number.
6. WAP to find sum of the following series for n terms: 1 – 2/2! + 3/3!----------n/n!

7. WAP to calculate the sum and product of two compatible matrices.

Section: B (Visual Python)

All the programs should be written using user defined functions, wherever possible.

1. Write a menu-driven program to create mathematical 3D objects
   I. curve
   II. sphere
   III. cone
   IV. arrow
   V. ring
   VI. Cylinder.
2. WAP to read n integers and display them as a histogram.
3. WAP to display sine, cosine, polynomial and exponential curves.
4. WAP to plot a graph of people with pulse rate p vs. height h. The values of p and h are to be entered by the user.
5. WAP to calculate the mass m in a chemical reaction. The mass m (in gms) disintegrates according to the formula m=60/(t+2), where t is the time in hours. Sketch a graph for t vs. m, where t>=0.
6. A population of 1000 bacteria is introduced into a nutrient medium. The population p grows as follows:
   P(t) = (15000(1+t))/(15+ *e*)
   where the time t is measured in hours. WAP to determine the size of the population at given time t and plot a graph for P vs t for the specified time interval.
7. Input initial velocity and acceleration, and plot the following graphs depicting equations of motion:
   I. velocity wrt time (v=u+at)
   II. distance wrt time (s=u*t+0.5*a*t*t)
   III. distance wrt velocity (s=(v*v-u*u)/2*a)