
Elective Subjects

PROBLEM SOLVING USING COMPUTERS(PYTHON PROGRAMMING)

1.1 Course Description

Provides fundamental knowledge of, and practical experience of the python.Subject also provides the various techniques of solving various computer problem.

1.2 Learning Targets/Outcomes

Upon successful completion of the course, the student will be able to:

- Analyze a specific problem situation requiring the development of a computer application
- Implement a computer application using wisely the elements of the python language;
- Can Make an application of reduced scale, but correct, modular, readable, and well documented ;

1.3 Course Organization

There will be 4-hours lecture per week, 2 hours Laboratory session each week. Please note that some of the tutorial/discussion sessions may be converted to lectures if need be. Students are expected to attend all lectures and to participate in class discussions.

2.1 Text Book

1. P. K. Sinha&PritiSinha , "Computer Fundamentals", BPB Publications, 2007.
2. T. Budd, Exploring Python, TMH, 1st Ed, 2011

2.2 Reference Book

- Dr. Anita Goel, Computer Fundamentals, Pearson Education, 2010.
- Python Tutorial/Documentation www.python.org2010
- Allen Downey, Jeffrey Elkner, Chris Meyers , How to think like a computer scientist

3. Teaching Schedule

3.1 Computer Fundamentals

3.1.1 Introduction to Computers:
3.1.2 Characteristics of Computers,
3.1.3 Uses of computers,
3.1.4 Types and generations of Computers.

3.2 Basic Computer Organization

3.2.1Units of a computer, CPU
3.2.2 ALU, memory hierarchy
3.2.3 Registers

3.2.4 I/O devices

3.3 Planning the Computer Program
3.3.1 Concept of problem solving
3.3.2 Problem definition, Program design,
3.3.3 Debugging,
3.3.4 Types of errors in programming, Documentation

3.4 Techniques of Problem Solving
3.4.1Flowcharting, decision table,
3.4.2 Algorithms,
3.4.3 Structured programming concepts,
3.4.4 Programming methodologies viz. top-down and bottom-up programming.

3.5Overview of Programming
3.5.1 Structure of a Python Program
3.5.2 Elements of Python
3.5.3 Introduction to Python:
3.5.3 Python Interpreter

3.6 Introduction to Python
3.6.1Using Python as calculator
3.6.2Python shell,
3.6.3Indentation.
3.6.4Atoms

3.7 Introduction to Python
3.7.1 Identifiers and keywords
3.7.2 Identifiers and keywords ….contd
3.7.3 Literals, Strings ….contd

3.8 Introduction to Python
3.8.1 Operators-Arithmetic operator, Relational
3.8.2 Logical or Boolean operator, Assignment Operator, Ternary operator,
3.8.3 Bit wise operator, Increment or Decrement operator

3.9 Creating Python Programs
3.9.1 Input and Output Statements
3.9.2 Input and Output Statements…Continue
3.9.3 Control statements (Looping-whileLoop)
3.9.4 Control statements (Looping-whileLoop)-Contimue

3.10 Creating Python Programs
3.10.1 For Loop
3.10.2 Loop Control Conditional Statement- if...else,
3.10.3 Difference between break,
3.10.4 Continue and pass

3.11 Structures
3.11.1 Numbers, Strings, Lists
3.11.2 Numbers, Strings, Lists…Continue
3.11.3 Tuples,
3.11.4 Dictionary,

3.12 Structures
3.12.1 Date & Time,
3.12.2 Date & Time…Continue
3.12.3 Modules
3.12.4 Modules---Continue

3.13 Structures
3.13.1 Defining Functions
3.13.2Defining Functions..Continue
3.13.3 Exit function
3.13.4Ddefault arguments.

3.14 **Introduction to Advanced Python**
3.14.1 Objects and Classes
3.14.2 Objects and Classes
3.14.3 Inheritance,
3.14.4 Inheritance --Continue

3.15 **Introduction to Advanced Python**
3.15.1 Regular Expressions
3.15.2 Event Driven Programming,
3.15.3 GUI Programming
3.15.4 GUI Programming..continue

4.Plagiarism
The University intends to develop and promote original work. Taking another person's words or ideas and using them as if they were your own or Plagiarism, as it is called, is taken very seriously at the University. Plagiarism may be deliberate or accidental.

Software Lab using Python:

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. velocitywrt time (v=u+at)
   II. distancewrt time (s=u*t+0.5*a*t*t)
   III. distancewrt velocity (s=(v*v-u*u)/2*a)