**Problem Solving and Programming**

**SEMESTER - I L-T-P-C : 3-1-3-5.5**

**Course Objectives:**

1. To introduce programming through Visual programming tool - Scratch
2. To teach problem solving through Flow charting tool - Raptor
3. To elucidate problem solving through python programming language
4. To introduce function-oriented programming paradigm through python
5. To train in development of solutions using modular concepts
6. To teach practical Pythonic solution patterns

**Unit – 1: Visual Programming through Scratch and App Inventor**

Introduction to programming concepts with scratch, Scratch environment, sprites looks and motion, Angles and directions, repetition and variation, changing costumes, adding background, Input/Output, variables and operators. Working with sounds and sprite communication and creating stories, App Generation.

**Learning Outcomes**: Student should be able to

1. Develop a program controlled by a loop. (L3)
2. Experiment with “costumes” to change the appearance of sprites.(L3)
3. Perform Input, Output Operations using scratch. (L3)
4. Perform computation using common mathematical formulas. (L3)
5. Develop programs by passing messages between sprites. (L3)
6. Build Mobile apps using App Inventor (L3)

**Text Book:**

<https://www.cse.msu.edu/~stockman/ITEC/Scratch/BGC2011Scratch-Rev1.pdf>

<https://nostarch.com/scratchplayground>

<http://fusecontent.education.vic.gov.au/9f79537a-66fc-4070-a5ce-e3aa315888a1/scratchreferenceguide14.pdf>

**Unit – 2: Flowchart design through Raptor**

Flow chart symbols, Input/Output, Assignment, operators, conditional if, repetition, function and sub charts. Example problems(section 1) – Finding maximum of 3 numbers, Unit converters, Interest calculators, multiplication tables, GCD of 2 numbers

Example problems(section 2) - Fibonacci generation, prime number generation. Minimum, Maximum and average of n numbers, Linear search, Binary Search.

**Learning outcomes:** Student should be able to

1. Select flowchart symbols for solving problems. (L1)
2. Develop basic flowcharts for performing Input, Output and Computations (L3)
3. Solve numerical problems using Raptor (L3)
4. Analyse problems by modular approach using Raptor (L4)

**Text Book:**

<https://raptor.martincarlisle.com/>

1. Download and Install Raptor software
2. Use the tool to draw flowcharts for the problems given.

**Unit – 3 : Introduction to Python**

Python – Numbers, Strings, Variables, operators, expressions, statements, String operations, Math function calls, Input/Output statements, Conditional If, while and for loops, User defined Functions, parameters to functions, recursive functions, Turtle Graphics.

**Learning outcomes**

1. Interpret numbers, strings, variables, operators, expressions and math functions using Python Interactive Mode. (L2)
2. Solve simple problems using control structures, input and output statements. (L3)
3. Develop user defined functions(recursive and non-recursive). (L3)
4. Build Python programs for section 1 raptor flowcharts. (L3)
5. Develop Python programs for creating various graphical shapes using turtle graphics. (L3)

Chapters 1 to 3

**Text Book:**

<http://www.ict.ru.ac.za/Resources/cspw/thinkcspy3/thinkcspy3.pdf>

**Unit – 4 : Data Structures and Idiomatic Programming in Python**

Lists, Tuples, Dictionaries, Strings, Files and their libraries. Beautiful Idiomatic approach to solve programming problems.

**Learning outcomes:** Student should be able to

1. Summarize the features of lists, tuples, dictionaries, strings and files. (L2)
2. Demonstrate best practices of “Beautiful Idiomatic Python”. (L2)
3. Build Python programs for section 2 raptor flowcharts. (L3).

<https://zhanxw.com/blog/wp-content/uploads/2013/03/BeautifulCode_2.pdf>

**Unit – 5 : Event driven Programming**

Turtle Bar Chart, Event Driven programming. Key press events, Mouse events, timer events.

Chapters 4 to 5 and Chapter 10

**Learning outcomes:** Student should be able to

1. Develop python programs to draw bar charts using turtle graphics. (L3)
2. Apply event driven programming and timers using Python (L3)

**Laboratory Experiments**

1. Design a script in Scratch to make a sprite to draw geometrical shapes such as Circle, Triangle, Square, Pentagon.
2. Design a script in Scratch to make a sprite to ask the user to enter two different numbers and an arithmetic operator and then calculate and display the result.
3. Design a Memory Game in Scratch which allows the user to identify positions of similar objects in a 3 x 3 matrix.
4. Construct flowcharts to
   1. calculate the maximum, minimum and average of N numbers
   2. develop a calculator to convert time, distance, area, volume and temperature from one unit to another.
5. Construct flowcharts with separate procedures to
   1. calculate simple and compound interest for various parameters specified by the user
   2. calculate the greatest common divisor using iteration and recursion for two numbers as specified by the user
6. Construct flowcharts with procedures to
   1. generate first N numbers in the Fibonacci series
   2. generate N Prime numbers
7. Design a flowchart to perform Linear search on list of N unsorted numbers(Iterative and recursive)
8. \*\*\* Design a flowchart to perform Binary search on list of N sorted numbers(Iterative and recursive)
9. Design a flowchart to determine the number of characters and lines in a text file specified by the user
10. Design a Python script to convert a Binary number to Decimal number and verify if it is a Perfect number.
11. Design a Python script to determine if a given string is a Palindrome using recursion
12. Design a Python script to sort numbers specified in a text file using lists.
13. \*\*\* Design a Python script to determine the difference in date for given two dates in YYYY:MM:DD format(0 <= YYYY <= 9999, 1 <= MM <= 12, 1 <= DD <= 31) following the leap year rules.
14. Design a Python Script to determine the Square Root of a given number without using inbuilt functions in Python.
15. \*\*\* Design a Python Script to determine the time difference between two given times in HH:MM:SS format.( 0 <= HH <= 23, 0 <= MM <= 59, 0 <= SS <= 59)
16. \*\*\* Design a Python Script to find the value of (Sine, Cosine, Log, PI, *e* ) of a given number using infinite series of the function.
17. \*\*\* Design a Python Script to convert a given number to words
18. \*\*\* Design a Python Script to convert a given number to roman number.
19. \*\*\* Design a Python Script to generate the frequency count of words in a text file.
20. \*\*\* Design a Python Script to print a spiral pattern for a 2 dimensional matrix.
21. \*\*\* Design a Python Script to implement Gaussian Elimination method.
22. \*\*\* Design a Python script to generate statistical reports(Minimum, Maximum, Count, Average, Sum etc) on public datasets.
23. \*\*\* Design a Python script using the Turtle graphics library to construct a turtle bar chart representing the grades obtained by N students read from a file categorising them into distinction, first class, second class, third class and failed.

**Text Book:**

<http://www.ict.ru.ac.za/Resources/cspw/thinkcspy3/thinkcspy3.pdf>

**Course outcomes:** Student should be able to

1. Create interactive visual programs using Scratch. (L4)
2. Develop flowcharts using raptor to solve the given problems. (L4)
3. Develop Python programs for numerical and text based problems (L4)
4. Develop graphics and event based programming using Python (L4)
5. Develop Python programs using beautiful Pythonic idiomatic practices (L4)