



Course Name : **Bachelor of Science**

Discipline : **Computer Science**

(Those who join in 2015 and after)

**Self Learning Courses:**

Subject	Credit	Ext =Tot	Subject Code
Microcontrollers and Embedded System Development in C	5	100 = 100	U1CSSL1
Introduction to Python Programming	5	100 = 100	U1CSSL2

**Microcontrollers and Embedded System Development in C**

**Credit: 5**

**Subject Code: U1CSSL1**

**Total marks 100**

**Study Objective:** This topic will motivate students to explore on their own about microcontroller programming basics and to design embedded system based applications.

**Prerequisite:** Knowledge on C language and computer organization.

Unit 1: 8051 Microcontrollers and Internal architecture

Microcontrollers & embedded processors, 8051 family overview, Inside 8051, Program counter and ROM, 8051 flag bits and the PSW register, 8051 register banks and stack.

Unit 2: 8051 Programming in C

Data types, Time delay in 8051, I/O programming in 8051, Logic operations in 8051, Data conversion programs in 8051, Accessing ROM and Data serialization using 8051.

Unit 3: Timer programming in C

8051 timers, programming timers in C, programming counters in C

Unit 4: Serial port programming in C

Basics of serial communication, 8051 connection to RS232, Serial port programming in C

Unit 5: Interrupts programming in C

8051 interrupts, Timer interrupts, external hardware interrupts, serial communication interrupts and all interrupt programming in C

**Text book:**

The 8051 Microcontroller and Embedded Systems – Using Assembly and C, Second Edition, Prentice Hall. Authors: *Muhammad Ali Mazidi, Janice Gillispie Mazidi & Rolin D. McKinlay.*

Chapters: 1, 2, 7,9,10 & 11. (Excluding Assembly language programming)

-----  
**SELF LEARNING COURSE -2**  
**INTRODUCTION TO PYTHON PROGRAMMING**

**Credit: 5**

**Subject Code: U1CSSL2**

**Total marks 100**

**Learning Objectives:**

1. To develop logic for Problem Solving.
2. To develop problem solving skills and their implementation through **Python (version: 2.7)**



3. To understand Open Source Software.

*Prerequisite: Knowledge on Java and C++ Programming.*

### **Unit I:**

**Getting Started:** Introduction to Python- an interpreted high level language, interactive mode and script mode. **Variables, Expressions and Statements:** Values, Variables and keywords; Operators and Operands in Python: (Arithmetic, relational and logical operators), operator precedence, Expressions and Statements (Assignment statement); Taking input (using raw\_input() and input()) and displaying output(print statement); Putting Comments.

### **Unit II:**

**Functions: Importing Modules (entire module or selected objects),** invoking built in functions, functions from math module (for e.g. ceil, floor, fabs, exp, log, log10, pow, sqrt, cos, sin, tan, degrees, radians), using random() and randint() functions of random module to generate random numbers, composition. **Defining functions,** invoking functions, passing parameters (*default parameter values, keyword arguments*), scope of variables, void functions and functions returning values, flow of execution.

### **Unit III:**

**Conditional constructs and looping:** if else statement While, For (range function), break, continue, else, pass, Nested loops, use of compound expression in conditional constructs and looping

### **Unit IV:**

**PROGRAMMING WITH PYTHON: Strings:** Creating, initialising and accessing the elements; String operators: +, \*, in, not in, range slice [n:m]; Comparing strings using relational operators; String functions & methods: len, capitalize, find, isalnum, isalpha, isdigit, lower, islower, isupper, upper, lstrip, rstrip, isspace, istitle, partition, replace, join, split, count, decode, encode, swapcase, Pattern Matching

### **Unit V:**

**Lists:** Concept of mutable lists, creating, initializing and accessing the elements, traversing, appending, updating and deleting elements; List operations (joining, list slices); List functions & methods: len, insert, append, extend, sort, remove, reverse, pop **Dictionaries:** Concept of key-value pair, creating, initializing and accessing the elements in a dictionary, traversing, appending, updating and deleting elements. Dictionary functions & Methods: cmp, len, clear(), get(), has\_key(), items(), keys(), update(), values(). **Tuples:** Immutable concept, creating, initialising and accessing the elements in a tuple; Tuple functions: cmp(), len(), max(), min(), tuple()

TEXTBOOKS:

1. Open Source: **Python (version: 2.7):** <https://www.python.org/about/gettingstarted/>
2. A Byte of Python – Swaroop CH (A free book on Programming with Python).