

Semester-II

Major/DS Course (Core)

COMP 2011: Programming Fundamentals using C

Credit: 03

45 Hours

Course Objective

The course is designed to provide complete knowledge of C language. Students will be able to develop logics which will help them to create programs, applications in C. By learning the basic programming constructs of C, they can easily switch over to any other language in future.

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- i. Develop problem solving skills coupled with top-down design principles.
- ii. Become skilled at developing simple algorithms and flow charts.
- iii. Convert the algorithms into simple C programs.
- iv. Understand code organization and functional hierarchical decomposition.
- v. Develop simple C programs for solving real life problems.

Syllabus

Theory:

Introduction: Basic Structure, Character sets, Keywords, Identifiers, Constants, Variables, Data Types, Program Structure. Operators: Arithmetic, Relational, Logical and Assignment; Increment, Decrement and Conditional, Operator Precedence and Associations; Assignment, Initialization, Expressions. Expression evaluation and type conversion. Formatted input and output, Conditional statements, Branching and looping, Array. (15 Hrs)

Functions – Arguments passing, Return values and their types, recursion. String handling with arrays, String handling functions, Enumerated data types. Structures. Arrays of structures. Arrays within structures, union (10 Hrs)

Pointers: Declaration and initialization, accessing variables through pointer arithmetic, Pointers and arrays, String, Pointer to Functions and Structures, Dynamic Storage Allocation. (15Hrs)

File handlings: Opening, Closing, I/O operations. (5 Hrs)

C language Practical:

Credit: 01

30 Hours

1. WAP to print the sum and product of digits of an integer.
2. WAP to reverse a number.
3. WAP to compute the sum of the first n terms of the following series $S = 1 + 1/2 + 1/3 + 1/4 + \dots$
4. WAP to compute the sum of the first n terms of the following series $S = 1 - 2 + 3 - 4 + 5 - \dots$
5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
6. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
7. WAP to compute the factors of a given number.
8. Write a macro that swaps two numbers. WAP to use it.

9. WAP to print a triangle of stars as follows (take number of lines from user):

*

10. WAP to perform following actions on an array entered by the user:

- i) Print the even-valued elements
- ii) Print the odd-valued elements
- iii) Calculate and print the sum and average of the elements of array
- iv) Print the maximum and minimum element of array
- v) Remove the duplicates from the array
- vi) Print the array in reverse order

The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.

11. WAP that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.

12. Write a program that swaps two numbers using pointers.

13. Write a program in which a function is passed address of two variables and then alter its contents.

14. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main() function.

15. Write a program to find sum of n elements entered by the user. To write this program, allocate memory dynamically using malloc() / calloc() functions or new operator.

16. Write a menu driven program to perform following operations on strings:

- a) Show address of each character in string
- b) Concatenate two strings without using strcat function.
- c) Concatenate two strings using strcat function.
- d) Compare two strings
- e) Calculate length of the string (use pointers)
- f) Convert all lowercase characters to uppercase
- g) Convert all uppercase characters to lowercase
- h) Calculate number of vowels
- i) Reverse the string

17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.

18. WAP to display Fibonacci series (i) using recursion, (ii) using iteration

19. WAP to calculate Factorial of a number (i) using recursion, (ii) using iteration

20. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion.

21. Create Matrix class using templates. Write a menu-driven program to perform following Matrix

operations (2-D array implementation):

22. Sum b) Difference c) Product d) Transpose
23. 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.
24. WAP to display the first n terms of Fibonacci series.
25. WAP to find factorial of the given number.
26. WAP to find sum of the following series for n terms: $1 - 2/2! + 3/3! - \dots - n/n!$
27. WAP to calculate the sum and product of two compatible matrices.

Reference Books

1. C Programming by K. R. N. & Ritchie, PHI
2. Programming through C by Richard Johnsonbaugh and Martin Kalin, Pearson Education
3. Programming in C, E. Balagurusamy, (Mcgraw Hill)

Skill Enhancement Course (SEC)

COMP 2051: System Administration and Maintenance (Practical)

Credit: 03

90 Hours

Course Objective

System administration is the field of work in which someone manages one or more systems, be they software, hardware, servers or workstations. Its goal is ensuring the systems are running efficiently and effectively.

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- i. Troubleshoot and fix issues that compromise system performance or access to an IT service.
- ii. Make regular system improvements, such as upgrades based on evolving end-user and business requirements.
- iii. Maintain common system and network administration tasks and practices and how to implement and maintain standard services like email, file sharing, DNS and similar tasks.

Syllabus

(Introduction to Linux/Unix)

15 hrs

- ☐ Basics of operating system, services,
- ☐ Installation and configuration, maintenance
- ☐ What is linux/unix Operating systems, Kernel, API, cli, gui,
- ☐ Difference between linux/unix and other operating systems
- ☐ Features and Architecture
- ☐ Linux features, advantages, disadvantages

(Introduction to Windows)

15 hrs

- ☐ Windows as operating system, history, versions.
- ☐ PC hardware, BIOS, Devices and drivers,
- ☐ Kernel Configuration and building
- ☐ Application installation, configuration and maintenance
- ☐ Server services and Client services

Linux Desktop tour. Configuring desktop environment and desktop settings, PC Assembly & Maintenance

15 hrs

Basic Commands: Terminal, shell, cat, ls, cd, date, cal, man, echo, pwd, mkdir, rm, rmdir ps, killPackage

Installation Synaptic package manager

15 hrs

Windows:

30 hrs

Creating users – Admin and regular.

Path of their personal files. Adding and changing passwords. Difference between workgroup and domain.

Concept of roles .user profiles – creating and roaming Concept of Active Directory. Creating active directory in windows.

Process and Disk management

Windows Task manager. File systems – NTFS, FAT.

Services Control Panel

C:/program Files, C:/system C:/windows

Add /remove new hardware (like printer), Add/remove new programmes. Network Administration

Ipconfig, Ping, tracert, route, hostname, net, netstat, who am I , Set manual IP address, check connectivity – ipv4, ipv6 Administrator Tools

Control Panel -> Administrative Tools

Computer Management, Local security Policy, Performance Monitor, Task Scheduler, Antivirus and firewall.

Misc Start->Accessories->System tools -> All options (Remote desktop, backup/restore etc.) LAN – Configuration, Switch, Router, sharing printer, files and folder over the network.

Reference Books

- i. The Practice of System and Network Administration , Thomas A. Limoncelli , Christina J. Hogan , Strata R. Chalup
- ii. Modern System Administration, Jennifer Davis
- iii. PC assembly and Installation, Nitish Kumar , Dr. T.H Sheikh
- iv. Windows Operating System Fundamentals, Crystal Panek (Sybex)

Minor Courses:
(For other allied discipline)

COMP 2021: Python Programming

Credit: 03

45 Hours

Course Objective

The course is designed to introduce programming concepts using Python to students. The course aims to develop structured as well as object-oriented programming skills using Python. The course also aims to achieve competence amongst its students to develop correct and efficient Python programs to solve real life problems.

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Develop, document, and debug modular Python programs of reasonable complexity.
2. Implement arrays and user defined functions in Python.
3. Solve real life problems of reasonable complexity using suitable and efficient programming constructs in Python.
4. Solve real life problems of reasonable complexity using the concepts of object-oriented programming in Python.

Syllabus

Theory:

Planning the Computer Program: Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation. (10 Hrs)

Techniques of Problem Solving: Flowcharting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming.(10 Hrs)

Overview of Programming: Structure of a Python Program, Elements of Python (5 Hrs)

Introduction to Python: Python Interpreter, Using Python as calculator, Python shell, Indentation. 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, List , Tuple , Set and Dictionary.(10 Hrs)

Creating Python Programs: Input and Output Statements, Control statements (Branching, Looping, Conditional Statement, Exit function, Difference between break, continue and pass.), Defining Functions, default arguments, functions, structures (10Hrs)

Reference Books

- 1) T. Budd, Exploring Python, TMH, 1st Ed, 2011
- 2) Python Tutorial/Documentation www.python.org -
- 3) Allen Downey, Jeffrey Elkner, Chris Meyers , How to think like a computer scientist :learning with Python , Freely available online.2012
- 4) <http://docs.python.org/3/tutorial/index.html>
- 5) <http://interactivepython.org/courselib/static/pythonds>
- 6) <http://www.ibiblio.org/g2swap/byteofpython/read/>

Software Lab Based on Python(Practical):

Credit :: 1

30 Hours

Section: A (Simple programs)

- Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon users choice.
- 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

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
 - iv. 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 $t \geq 0$.
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:
 - velocity wrt time ($v=u+at$)
 - distance wrt time ($s=u*t+0.5*a*t*t$)
 - distance wrt velocity ($s=(v*v-u*u)/2*a$)

Multi/Interdisciplinary courses

(For Other discipline)

Semester-II

COMP 2031: Introduction to Internet

Credit: 03

45 Hours

Course Objective

This course is intended to teach the basics involved in publishing content on the World Wide Web. This includes the 'language of the Web' – HTML and the fundamental principles of how the Internet and the Web function.

Course Learning Outcomes

On successful completion of the course, students will be able to:

- i. Discuss elementary Internet concepts and history.
- ii. Make a successful Internet connection.
- iii. Demonstrate simple principles of Internet Protocol (IP) addressing.
- iv. Use and customize a web browser.
- v. Comprehend the basics of the internet and web terminologies.

Syllabus

Introduction : Evolution of Internet, concept of Intranet and Internet, Applications of Internet, Types of Connectivity such as dial – up, leased, VSAT. etc., Internet Server and Clients module in various Operating Systems.(5 Hrs)

Usenet and Internet Relay Chart Introduction to World Wide Web: Evolution of WWW, Basics Features, WWW Browsers, WWW servers, HTTP & URL's. (5 Hrs)

WWW Browsers: Basic features, Bookmarks, history. Progress indicators, Personalization of Browsers, Printing displayed pages and forms, Saving Web pages, Netscape Communicators, Internet Explorer, Search and Downloads.(5 Hrs)

Search Engines: Technology overview, Popular Search Engines.
How to register a website in search engine. (5 Hrs)

Internet Security: Overview of Internet Security threats, Firewalls, Introduction to AAA (5 Hrs)

HTML: (20 Hrs)

- **Unit-I: Introduction**
- **Unit-II: The Basics**
 - o The Head, the Body
 - o Colors, Attributes
 - o Lists, ordered and unordered
- **Unit-III: Links**
 - o Introduction
 - o Relative Links, Absolute Links
 - o Link Attributes

- o Using the ID Attribute to Link Within a Document

- **Unit-IV: Images**

- o Putting an Image on a Page
- o Using Images as Links
- o Putting an Image in the Background

- **Unit V: – Tables**

- o Creating a Table
- o Table Headers
- o Captions
- o Spanning Multiple Columns
- o Styling Table

Reference Books

1. Internetworking with TCP/IP – by D.E.Comer, PHI
2. Introduction to HTML and CSS -- O'Reilly