DETAILED SYLLABUS

Semester III

Sl. No.	Category	Code No.	Course Title	Hours per week			Total Contact Hrs/Week	Credit
				L	T	P		
1	Programme corecourse-1	CSPC301	Computer Programming	3	0	0	3	3
2	Programme corecourse-2	CSPC302	Computer System Organization	3	0	0	3	3
3	Programme corecourse-3	CSPC303	Operating Systems	2	0	0	2	2
4	Programme corecourse-4	CSPC304	Scripting Languages (Python /Perl – any one)	2	0	0	2	2
5	Programme corecourse-5	CSPC305	Digital Logic Design & Microprocessor	2	0	0	2	2
6	Programme corecourse-6	CSPC306	Computer Programming Lab	0	0	4	4	2
7	Programme corecourse-7	CSPC307	Scripting Languages Lab	0	0	4	4	2
8	Programme corecourse-8	CSPC308	Operating Systems Lab	0	0	2	2	1
9	Programme corecourse-9	CSPC309	Digital Logic Design & Microprocessor Lab	0	0	2	2	1
10	Summer Internship-I (4 weeks) after IInd Semester)	CSSI310	Summer Internship-I	0	0	0	0	2
			Total				**24**	**20**

Computer Programming

Course Code	CSPC301
Course Title	Computer Programming
Number of Credits	3(L : 3, T : 0, P : 0)
Prerequisites	Interaction with DOS / Windows Operating System
Course Category	Programme Core Course

Course Outcomes: -

After completion of the course student will be able to

1) Learn the common units of programming languages
2) Identify problems to be solved
3) Develop structured solutions to problems
4) Express solution in a machine readable form or a programming language

Detailed Course Contents

Module- 1: Introduction of Programming

Number of class hours: 8

Suggestive Learning Outcomes:

1) Identify problems to be solved
2) Understand Structure of a program
3) Learn various basic units of program and use them.

Detailed content of the unit:

1.1. Introduction to Problem Solving
1.2. Structured Language and an overview of C
1.3. Character set, Tokens, Constants, Variables
1.4. Key words and Identifiers
1.5. Data types used in C & their size.

Module- 2: Operators and I/O

Number of class hours: 4

Suggestive Learning Outcomes:

1) Explain and use various operators
2) Illustrate I/O operation

Detailed content of the unit:

2.1. Arithmetic, Relational, Logical and Bitwise Operators
2.2. Operator precedence

2.3. Input, Output, Formatting and File I/O

Module-3: Control statements

Number of class hours: 7

Suggestive Learning Outcomes:

1) Understand and learn various control statements
2) Recognize and apply control statements where ever required

Detailed content of the unit:

3.1. Decision making and branching statements
3.2. if statement (if, ifelse,else-if ladder, nested if-else)
3.3. Switch case statement.
3.4. Iterative/Loop statement
3.5. while, do-while
3.6. for Loop structure
3.7. Break and continue statement
3.8. Conditional and unconditional Goto statement

Module-4: Functions and Arrays

Number of class hours: 8

Suggestive Learning Outcomes:

1) Learn and identify types of functions and arrays
2) Apply them in solving problems

Detailed content of the unit:

4.1. Functions, Need of functions, Difference between library function and user defined Function.
4.2. Prototype declaration, Defining functions, Passing parameter types, Function call, Return values
4.3. Category of function (No argument No return value, No argument with return value, Argument with return value)
4.4. Advantages of arrays
4.5. Declaration and initialization of one dimensional, two dimensional and character arrays
4.6. Accessing array elements.

Module- 5: Recursion

Number of class hours: 3

Suggestive Learning Outcomes:

1) Illustrate recursive function
2) Use recursive function

Detailed content of the unit:

5.1. Recursion and use of memory stack
5.2. Types of recursion,
5.3. Advantages and disadvantages of recursive function.

References:

1. Programming in ANSI C, E. Balagurusamy, Tata McGraw-Hill
2. Outline of Programming with C, Byron Gottfried, Schaum, McGraw-Hill
3. Let Us C, Yashavant Kanetkar
4. Programming in C, Reema Thareja, OUP India
5. Problem Solving and Programming in C, R.S. Salaria, Khanna Publishing House
6. Programming in ANSI C, E. Balagurusamy, Tata McGraw-Hill
7. C Programming & Data Structures, B. A. Fouruzan and R. F. Gilberg, CENGAGE Learning.

****

Computer System Organisation

Course Code	CSPC302
Course Title	Computer System organisation
Number of Credits	3(L: 3, T:0, P:0)
Prerequisites	Knowledge of Number System
Course Category	Programme Core Course

Course Outcomes: -

After Completion of the course students will be able to:

1) Have a thorough understanding of functioning of digital computer system as such and its various subcomponents.
2) Understand computing requirement for a specific purpose
3) Analyse performance bottlenecks of the computing device
4) Choose appropriate computing device for a given use case.

Detailed Course Contents

Module- 1: Introduction to Computer System

Number of class hours: 6

Suggestive Learning Outcomes:

4) Identify and explain functionality of various parts of digital computer
5) Illustrate data representation.
6) Explain register and arithmatic microoperation.

Detailed content of the unit:

1.1. Structure of Computers: Computer Functional units, Von-Neumann architecture, Bus Structures.
1.2. Basic Operational Concepts, Data representation (Fixed and Floating point)
1.3. Error detecting codes.
1.4. Register Transfer and Micro Operations: Register transfer, Bus and memory transfers.
1.5. Basics of Arithmetic micro-operations, Logic micro-operations, Shift micro-operations, and Arithmetic logic shift unit.

Module- 2: Micro Programmed Control and Pipeline

Number of class hours: 8

Suggestive Learning Outcomes:

3) Explain working of Control unit.
4) Solve arithmetic problems
5) Understand concept of pipelining

Detailed content of the unit:

2.1. Micro Programmed Control: Control memory, Address sequencing, and design of control unit
2.2. Computer Arithmetic: Addition and Subtraction, Multiplication and Division algorithms, Floating-point arithmetic operation
2.3. Introduction to Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline Vector Processing, Array Processors

Module-3: Microprocessor

Number of class hours: 6

Suggestive Learning Outcomes:

3) Understand and learn various various parts of microprocessor
4) Recognize and apply various instructions .

Detailed content of the unit:

3.1. Introduction to Microproccessor Architecture
3.2. Instruction Set Architecture design principles fromprogrammer's perspective. One example microprocessor (Intel, ARM, etc).

Module-4: Programming the Basic Computer

Number of class hours: 5

Suggestive Learning Outcomes:

1) Write simple assembly level programs
2) Understand and explain procedure and macros
3) Evaluate arithmatic expression

Detailed content of the unit:

4.1. Assembly Language Programming: Simple programs, Assembly language programs involving logical, branch and call instructions
4.2. Sorting, evaluation of arithmetic expressions
4.3. String manipulation, assembler directives
4.4. Procedures and macros.

Module- 5: Memory and I/O Organization

Number of class hours: 5

Suggestive Learning Outcomes:

1) Understand and illustrate memory and I/O interfacing
2) Demonstarte various modes of I/O operation

Detailed content of the unit:

5.1. Memory and Digital Interfacing: addressing and address decoding
5.2. Interfacing RAM, ROM, EPROM, programmable peripheral interface
5.3. Various modes of operation and interfacing to processor, interfacing keyboard, displays etc.

References:

1. Computer System Architecture, M. Moris Mano, Pearson/PHI, India.
2. Microprocessors Interface, Douglas V.Hall, Tata McGraw-Hill.
3. Computer Organization, Carl Hamacher, Zvonks Vranesic, SafeaZaky, McGraw-Hill
4. Advanced Microprocessors and Peripherals- Architecture, Programming and interfacing, A.K.Ray, K.M.Bhurchandi, Tata McGraw-Hill, New Delhi, India.
5. Computer Organization and Design: A Hardwar/Software Interface (MIPS Edition) by Patterson and Hennessy

****

Operating System

Course Code	CSPC303
Course Title	Operating System
Number of Credits	2 (L: 2, T: 0, P: 0)
Prerequisites	Computer Organization and Digital Design
Course Category	Programme Core Course

Course Outcome:-

After completion of the course, students will be able to:

C.O. 1: Understand the basics of operating systems like kernel, shell, types and services of operating systems. (K2)

C.O. 2:Understand the concept of program, process and thread and analyse various CPU scheduling Algorithms. (K2,K3)

C.O. 3: Describe and analyse the memory management and its allocation policies. (K3)

C.O. 4: Understand the issues related to file system interface and implementation.(K2,K3)

C.O. 5: Understand disk management and explain disk scheduling algorithms for better utilization of external memory. (K2, K3)

C.O. 6: Configure OS in an efficient and secure manner. (K3)

Course Content:

Module 1 – Introduction

Learning Outcomes:-

Students will be able to-

1. Define an operating system.
2. Discuss history of operating system.
3. Discus about various types of operating systems and operating system services.
4. Define system call with an example.
5. Explain single, multi user operating system structure.

Content:

Overview of Operating System, basic concepts, UNIX/LINUX Architecture, Kernel, services and systems calls, system programs.

Module 2 – **Process Management and Memory management**

Learning Outcomes:-

Students will be able to-

1. Define process, threads and multithreading.
2. Understand b) process state diagram c) process control block.
3. Describe process creation and termination.
4. Explain various scheduling algorithms – FCFS, SJF, Priority, Round Robin,
5. Explain inter process communication.
6. Explain single partition allocation and multiple partition allocation, paging and segmentation.
7. Describe page replacement algorithms - FIFO, LRU, Optimal.
8. Define concept of thrashing and page fault.

Content:

Process Management: Process concepts, operations on processes, IPC, Process Scheduling, Multithreaded programming.

Memory management: Memory allocation, Swapping, Paging, Segmentation, Virtual Memory, various faults.

Module 3 –File management

Learning Outcomes:-

Students will be able to-

1. Define file management.
2. List and explain various file operations and file access methods.
3. Explain directory structure organization.
4. Describe the concept of file protection.
5. Understand Different types of file systems.

.

Content:

Concept of a file, access methods, directory structure, file system mounting, file sharing and protection, file system structure and implementation, directory implementation, free space management, efficiency and performance. Different types of file systems.

Module 4 – I/O System

Learning Outcomes:-

Students will be able to-

1. Explain disk structure.
2. Understand swap space management.
3. Explain various disk scheduling algorithms- FCFS, SST, SCAN,C-SCAN, LOOK.
4. Explain various RAID levels.

Content:
Mass storage structure - overview, disk structure, disk attachment, disk scheduling algorithms, swap space management, RAID types.

Module 5 –OS Security

Learning Outcomes:-

Students will be able to-

1. Understand and identify potential threats to operating system,

2.Explain different Authentication schemes.

2. Explain security features design to guard against threats.

Content:
Authentication, Access Control, Access Rights, System Logs

References/ Suggested Learning Resources:-

1. Operating System Concepts, Silberschatz and Galvin, Wiley India Limited
2. UNIX Concepts and Applications, Sumitabha Das, McGraw-Hill Education
3. Operating Systems, Internals and Design Principles, Stallings, Pearson Education, India
4. Operating System Concepts, Ekta Walia, Khanna Publishing House
5. Modern Operating Systems, Andrew S. Tanenbaum, Prentice Hall of India
6. Operating systems, Deitel & Deitel, Pearson Education, India
7.Principles of Operating Systems, Naresh Chauhan, Oxford University Press India.

Websites for Reference: http://swayam.gov.in

SCRIPTING LANGUAGE(PYTHON)

Course Code	CSPC304
Course Title	Scripting Language (Python)
Number of Credits	2 (L: 2, T: 0, P: 0)
Prerequisites	Knowledge of Programming
Course Category	Programme Core Course

<u>Course Outcomes: -</u>

After the completion of the course Student will be able to

1) Draw flow charts for solving different problems, develop efficient algorithms for solving a problem. (**K3**)
2) Use the various constructs of Python viz. conditional, iteration(**K1**)
3) Write programs making judicious use of Lists, Strings, Tuples, Dictionaries wherever required(**K3)**
4) Manage data using NumPy (**K3**)
5) Handle files and Modules in Python (**K2**)

<u>Course Content:-</u>

Module- 1: Introduction to Programming, Algorithm and Flowcharts

Number of class hours: 04 Hrs

Suggestive Learning Outcomes:

1) Understand the concept and evolution of Programming. **(K1)**
2) Understand the concepts and purposes of algorithm and flowchart. **(K1)**
3) Use algorithm and flowchart to solve problem independent of language. **(K3)**
4) Gain knowledge of different constructs of algorithm and flowchart. **(K2)**

Detailed content of the unit: - The basic Model of computation, Algorithms, Flowcharts, Programming Languages, Compilation, Testing & debugging and documentation, Flow Chart Symbols, Basic algorithms/flowcharts for sequential processing, Decision based processing and Iterative processing.

Module- 2: Introduction to Python

Number of class hours: 05 Hrs

Suggestive Learning Outcomes:

1) Understand features of Python that make it one the most popular languages in the industry. **(K1)**
2) Understand structure of Python problem. **(K2)**
3) Understand the areas where Python is used. **(K2)**

Detailed content of the unit: - Python Introduction, Technical Strength of Python, Introduction to Python Interpreter and program execution, Using Comments, Literals, Constants, Python's Built-in Data types, Numbers (Integers, Floats, Complex Numbers, Real, Sets), Strings (Slicing, Indexing, Concatenation, other operations on Strings), Accepting input from Console, printing statements, Simple 'Python' programs.

Module- 3: Operators, Expressions and Python Statements, Sequence data types

Number of class hours: 8 Hrs

Suggestive Learning Outcomes:

1) Use the basic operators and expressions available in Python in developing program. **(K3)**
2) Understand and use various Python statements like conditional constructs, looping constructs in writing Python program. **(K3)**
3) Work with various built-in Sequence datatypes and their use. **(K3)**
4) Understand the concept of mutable and immutable objects. **(K2)**

Detailed content of the unit: - Assignment statement, expressions, Arithmetic, Relational, Logical, Bitwise operators and their precedence, Conditional statements: if, if-else, if-elseif-else; simple programs, Notion of iterative computation and control flow –range function, While Statement, For loop, break statement, Continue Statement, Pass statement, else, assert.
Sequence Data Types: Lists, tuples and dictionary, (Slicing, Indexing, Concatenation, other operations on Sequence datatype), concept of mutability, Examples to include finding the maximum, minimum, mean; linear search on list/tuple of numbers, and counting the frequency of elements in a list using a dictionary.

Module- 4: Functions, File Processing, Modules

Number of class hours: 8 Hrs

Suggestive Learning Outcomes:

1) Apply the in-built functions available in Python in solving different problems. **(K3)**
2) Work with modular approach using user defined functions. **(K2)**
3) Work with files and reading /writing onto files. **(K3)**
4) Understand the concept of modules and importing, loading and reloading of modules in programs. **(K1)**

Detailed content of the unit: - Functions Top-down approach of problem solving, Modular programming and functions, Function parameters, Local variables, the Return statement, DocStrings, Global statement, Default argument values, Keyword arguments, VarArgs parameters. Library functions, Time functions, Recursion, Concept of Files, File opening in various modes and closing of a file, Reading from a file, Writing onto a file, File functions,
Command Line arguments, Scope of objects and Names, LEGB Rule, Module Basics, Module Files as Namespaces, Import Model, Reloading Modules.

Module- 5: NumPy Basics

Number of class hours: 05 Hrs

Suggestive Learning Outcomes:

1) Work on NumPy array manipulation to access data and subarrays and to split, reshape, join arrays etc. **(K3)**

Detailed content of the unit: - Introduction to NumPy, ndarray, datatypes, array attributes, array creation routines, Array from Existing Data, Array from Numerical Ranges, Indexing & Slicing.

References: - 1) Python Programming- A modular Approach (with Graphics, database, Mobile and Web Applications by Sheetal Taneja and Naveen Kumar, Pearson.

2) Head First Python by Paul Berry, O'Reilly

3) Dive into Python by Mark Pilgrim, APress

4) Beginning Programming with Python Dummies by John Paul Meuller.

5) Programming and Problem Solving Through Python Language, Prof. Satish Jain, Shashi Singh, BPB Publication.

OR

SCRIPTING LANGUAGE (PERL)

Course Code	CSPC304
Course Title	Scripting Language(Perl)
Number of Credits	2 (L: 2, T: 0, P: 0)
Prerequisites	Knowledge of Programming and Linux
Course Category	Programme Core Course

Course Outcomes: -

After the completion of the course Student will be able to

1) The fundamentals to create functional Perl scripts **(K1)**.
2) Know about data structures, flow control mechanisms, regular expressions, and subroutines and modules. **(K2)**
3) Know about the usage of Database Access using Perl. **(K3)**

Course Content:-

Module- 1: An Overview of Perl

Number of class hours: 08 Hrs

Suggestive Learning Outcomes:

1) Understand features and structure of Perl. **(K1)**
2) Understand the areas where Perl is used. **(K1)**
3) Understand the uses of Filehandles, Operators, Control Structures, Regular Expressions. **(K3)**

Detailed content of the unit: - Perl Introduction, Perl Environment, Perl Installation, Natural and Artificial Languages, A Grade Example, Filehandles, Operators, Control Structures, Regular Expressions, List Processing

Module- 2: The Gory Details

Number of class hours: 8 Hrs

Suggestive Learning Outcomes:

1) Use the basic operators and expressions available in Perl in developing program. **(K2)**
2) Understand and use various subroutines, formats in writing Perl program. **(K3)**
3) Work with various built-in datatypes and their use **(K3)**
4) Understand the concept of pattern matching, variables and hashes. **(K2)**

Detailed content of the unit: - Lexical Texture, Built-in Data Types, Terms, Pattern Matching, Operators, Statements and Declarations, Subroutines, Formats, Special Variables, Hashes

Module- 3: References and Nested Data Structures

Number of class hours: 05 Hrs

Suggestive Learning Outcomes:

1) Understand the concept of References, Braces, Brackets, and Quoting, **(K2)**
2) Understand the concept of lists and data structure codes. **(K3)**

Detailed content of the unit: - About Reference, Creating Hard References, Using Hard References, Symbolic References, Braces, Brackets, and Quoting, Manipulating Lists of Lists, Data Structure Code Examples.

Module- 4: Functions, Packages, Modules, and Object Classes

Number of class hours: 05 Hrs

Suggestive Learning Outcomes:

1) Work on Understand the concept of Functions. **(K2)**
2) Understand the concept of Modules. **(K2)**
3) Understand the concept of Object, Instance variables. **(K1)**
4) Understand the concept of Packages. **(K1)**

Detailed content of the unit: - Perl Functions by Category, Perl Functions in Alphabetical Order, Packages, Modules, Objects, Perl's Objects, Brief Refresher on Object-Oriented Programming, Using Tied Variables, About Object Design, Instance Variables, Containment, Implementation, Delegation,

Module- 4: Database Access

Number of class hours: 05 Hrs

Suggestive Learning Outcomes:

1) Understand the concept of Database file. **(K3)**
2) Understand the concept of SQL Command Using DBI and DBD. **(K3)**

Detailed content of the unit: - Making and Using a DBM File, Emptying a DBM File, Converting Between DBM Files, Merging DBM Files, Locking DBM Files, Sorting Large DBM Files, Executing an SQL Command Using DBI and DBD

References: -

1) Programming Perl by Larry Wall, Tom Christiansen, & Randal Schwartz; O'REILLY.
2) Learning Perl by Tom Phoenix, Randal L. Schwartz, O'REILLY.
3) Perl Cookbook by Tom Christiansen, Nathan Torkington, O'REILLY.
4) Perl: The Complete Reference by Martin C Brown, McGraw-Hill

****

Digital Logic Design & Microprocessor

Course Code	CSPC305
Course Title	Digital logic design & Microprocessor
Number of Credits	2 (L: 2, T:0, P:0)
Prerequisites	--------
Course Category	Programme Core Course

Course Outcomes: -

After Completion of the course students will be able to:

1) Test the digital systems, logic families and logic gates.
2) Construct combinational logical circuit.

3) Construct sequential logical circuit.
4) Use registers and instructions of 8086.
5) Develop assembly language programs using 8086.

Detailed Course Contents

Module- 1: Number systems, Digital Logic families and Logic Gates

Number of class hours: 4 to 5.

Suggestive Learning Outcomes:

1) Convert the number from the given number system to the specified number system.
2) Perform the given binary arithmetic operation on the given data.
3) Describe the characteristics of the given Digital Logic family.
4) Derive the truth table of the given basic logic gate/derived logic gate.
5) Apply Boolean algebra for designing the given logic circuit.
6) Design the logical circuit for the given application.

Detailed content of the unit:

1. Terms - Bit, Byte, Nibble.
2. Number systems – Decimal, Binary, Octal and Hexadecimal and their conversions from one number system to another (Integer and fractional).
3. Codes – BCD, Gray, ASCII, EBCDIC.
4. Binary arithmetic – Compliments – 1’s and 2’s, Addition, subtraction, multiplication and division. (up to 8 bit).
5. Applications of digital circuits, Comparison of TTL, CMOS, ECL, Characteristics of digital ICs. (Propagation Delay, Noise Margin, Power dissipation, Fan-in Fan-out, Threshold logic levels).
6. Basic gates (AND, OR, NOT), Derived gates (NAND, NOR, EX-OR, EX-NOR), Universal gates.
7. Basic logic operations using laws of Boolean algebra. DE Morgan’s Theorems.

Module- 2: Combinational Logic Circuits

Number of class hours: 5 to 6.

Suggestive Learning Outcomes:

1) Simplify the given logical expression using Sum-of-Product (SOP) and Product-of-Sum (POS) approaches.
2) Minimize the given logical function using Karnaugh’s map (K-MAP).
3) Explain the approach of designing the given data for half-adder/half-subtractor using K-MAP.
4) Construct the logical diagrams of multiplexer/demultiplexer to solve the given problem.

Detailed content of the unit:

1. Standard/canonical forms for Boolean functions, Min terms and Max terms.
2. Simplification of logical circuit by way of SOP and POS approaches.
3. Expression simplification using Boolean algebra techniques (i.e. K-MAP of 2,3,4 variable K-MAPs).
4. Construction of Half Adder and Half subtractor using K-MAP.
5. Necessity, principle and types of multiplexers and demultiplexer.

Module- 3:Sequential Logic Circuits

Number of class hours: 5 to 6.

Suggestive Learning Outcomes:

1) Differentiate characteristics of the given logic circuits.
2) Identify the given situations where edge triggering is preferred over level triggering with justification.
3) Explain with sketches the working principle of given type of flip flop.
4) Choose relevant type of flip-flop (SR/JK/D/T) based on given number of inputs and the manner in which they affect the binary state of flip-flops.

Detailed content of the unit:

1. Combinational and sequential logic Circuits. Block diagram of sequential circuit.
2. Flip-flops: One-it memory cell symbol applications of flip-flops, Types of triggering flip-flops, Edge triggered and level triggered.
3. Principle of working of different flip-flop types-SR, JK, D and T Flip-flop.

Module- 4:Microprocessor: 8086 and Model Microprocessors

Number of class hours: 6 to 7.

Suggestive Learning Outcomes:

1) Explain the process of executing the given instructions in 8086 microprocessors.
2) Derive physical address to locate the given data from memory segmentation.
3) Compare architecture of Microprocessor 8086 with the specified processor on the given parameters.
4) Select the CISC/RISC architecture-based processor for the given situation with justification.

Detailed content of the unit:

1. Evolution of Microprocessor and type. 16-bit Microprocessor-8086.

2. Features of 8086, pin diagram and architecture of 8086, Flag register and segment registers of 8086, Minimum mode and maximum mode of operation. Timing diagram. Concept of memory segment and pipelining, physical address generation.
3. Overview of Pentium family and processors.
4. Characteristics of RISC processor.
5. CISC with RISC in terms of Instruction set, Length, addressing modes.

Module- 5:Assembly Language Programming using 8086.

Number of class hours: 6 to 7.

Suggestive Learning Outcomes:

1) Select relevant addressing mode of 8086 to identify instruction for solving the given problem with justification.
2) Choose relevant instruction to perform the given operation from the instruction set of 8086 with justification.
3) Develop assembly language programme to solve the given expression.
4) Develop the assembly language program to solve the given problem using decision making and looping structure.

Detailed content of the unit:

1. Programming model of 8086 assembly language program.
2. Addressing modes of 8086 with examples.
3. Group of instruction set-Data transfer, Arithmetic and Logical, Branch and loop, Flag manipulation, shift and rotate and string instructions, (only format and examples).
4. Assembly Language programs for (8-bit and 16-bit) Addition, Subtraction, Multiplication, Division, Decision making and looping.

References:

Sl.no.	**Title of Book**	**Author(s)**	**Publication**
1	Modern Digital Electronics	R.P.Jain	McGraw Hill Education, New Delhi.
2	Digital Principles and Applications	Donald P. Leach, Albert Paul Malvino, Gautam Saha.	McGraw Hill Education, New Delhi.
3	Advanced Microprocessor and Peripherals 3/E	K. M. Bhurchandi, A.K. Roy	McGraw Hill Education, New Delhi.
4	8086 Programming and advance processor architecture	M.T. Savaliya	Wiley India, New Delhi

Suggested software/Learning Websites:

1. http://www.electrical4u.com/some-common-applications-of-logic-gates/
2. http://www.zeepedia.com
3. http://www.cburch.com/logisim
4. http://www.logiccircuit.org/download.html
5. http://www.learnabout-electronics.org
6. http://www.firmcodes.com/different-risc-sics-architecture/
7. http://www.arm.com
8. http://meseee.ce.rit.edu/551-projects/fall2012/1-1.pdf
9. http://www.intel.com

****

Computer Programming Lab

Course Code	CSPC306
Course Title	Computer Programming Lab
Number of Credits	2 (L : 0, T : 0, P : 4)
Prerequisites	Interaction with DOS / Windows Operating System
Course Category	Programme Core Courses

Course Outcomes: -

After completion of the course student will be able to

1) Handle of Computer System properly.
2) Apply different logics to solve given problem.
3) Understand different steps and stages to develop complex program
4) Write program using different implementations for the same problem.
5) Identify different types of errors as syntax, semantic, fatal, linker & logical.
6) Debugging of programs.

Detailed Course Contents

Module No.	List of suggested programs/ experiments	No. of class hour
1	1. Display Hello World 2. Taking input from user 3. Find ASCII value of Character 4. Use of gets() function	4

2	5. Displaying hexadecimal, decimal, octal number format of the entered numbers. 6. Displaying entered number with leading zeros and trailing zeros. 7. Displaying entered number with right and left justification. 8. Displaying with different formatting specifiers. 9. Swapping two numbers 10. To find greatest / smallest of three numbers. 11. To display pass class, second-class, distinction according to the marks entered from the keyboard. 12. To find even or odd numbers. 13. To display spellings of number 1-10 on entry. 14. Implementation and displaying the menu to execute 1. ADD, 2. 15. SUBTRACT 3. MULTIPLICATION, 4. DIVISION using switch case. 16. Handling with unformatted, formatted files in different 17. operational mode.	10
3	18. To display our College name twenty times on screen. 19. To demonstrate Continue and Break statements within loop 20. structure. 21. To add first ‘n’ natural, even, odd numbers using different loop 22. structures. 23. To find GCD, LCM of two integral numbers. 24. To generate simple number triangle for n rows. 25. To generate Pascal triangle for n rows. 26. To add the series 1 + (1 + 2) + (1 + 2 + 3) + …+ (1 +2 +3 + …+n) 27. To generate all prime numbers within the given range. 28. To find all the Armstrong numbers within 100 to 1000.	10
4	29. Display elements of array 30. Reverse an Array 31. Insert element to array 32. Find largest and smallest element in Array 33. Display teo dimentional array. 34. Addition and subtraction of Matrix 35. To calculate multiplication of 2-dimensional matrix. 36. To find the number of vowels and consonants in a string. 37. Implementation of strlen(), strcpy(), strcat() and strcmp() functions. 38. To check whether a string is palindrome or not. 39. Use of all types of functions	10
5	40. Using recursion write program 41. To calculate sum of two numbers 42. To calculate factorial of any given number. 43. Display Fibonacci series 44. Reverse a string 45. Sum of Digits	6

References:

1. Programming in ANSI C, E. Balagurusamy, Tata McGraw-Hill
2. Outline of Programming with C, Byron Gottfried, Schaum, McGraw-Hill
3. Let Us C, Yashavant Kanetkar
4. Programming in C, Reema Thareja, OUP India
5. Problem Solving and Programming in C, R.S. Salaria, Khanna Publishing House
6. Programming in ANSI C, E. Balagurusamy, Tata McGraw-Hill
7. C Programming & Data Structures, B. A. Fouruzan and R. F. Gilberg, CENGAGE Learning.

****

SCRIPTING LANGUAGE LAB(PYTHON)

Course Code	CSPC307
Course Title	Scripting Language Lab(Python)
Number of Credits	2 (L: 0, T: 0, P: 4)
Prerequisites	Knowledge of Programming
Course Category	Programme Core Course

Course Outcomes: -

After the completion of the course Student will be able to

1) Draw flow charts for solving different problems, develop efficient algorithms for solving a problem. (**K3**)
2) Use the various constructs of Python viz. conditional, iteration(**K3**)
3) Write programs making judicious use of Lists, Strings, Tuples, Dictionaries wherever required(**K3)**
4) Manage data using NumPy (**K3**)

Practical Assignments:

1) Write a program to print all Armstrong numbers in a given range. Note: An Armstrong number is a number whose sum of cubes of digits is equal to the number itself. E.g. 370=33+73+03
2) Write a function to obtain sum n terms of the following series for any positive integer value of X
   X +X3 /3! +X5 /5! ! +X7 /7! + …
3) Write a function to obtain sum n terms of the following series for any positive integer value of X
   1+x/1!+x2/2!+x3/3!+…
4) Write a program to multiply two numbers by repeated addition e.g.
   6*7 = 6+6+6+6+6+6+6
5) Write a program to compute the wages of a daily labourer as per the following rules :-

Hours Worked Rate Applicable Upto first 8 hrs Rs100/-

a) For next 4 hrs Rs30/- per hr extra

b) For next 4 hrs Rs40/- per hr extra

c) For next 4 hrs Rs50/- per hr extra

d) For rest Rs60/- per hr extra

6) Accept the name of the labourer and no. of hours worked. Calculate and display the wages. The program should run for N number of labourers as specified by the user.

7) Write a function that takes a string as parameter and returns a string with every successive repetitive character replaced by? e.g. school may become school.

8) Write a program that takes in a sentence as input and displays the number of words, number of capital letters, no. of small letters and number of special symbols.

9) Write a Python program that takes list of numbers as input from the user and produces a cumulative list where each element in the list at any position n is sum of all elements at positions upto n-1.

10) Write a program which takes list of numbers as input and finds:

a) The largest number in the list

b) The smallest number in the list

c) Product of all the items in the list

11) Write a Python function that takes two lists and returns True if they have at least one common item.

12) Write a Python program to combine two dictionary adding values for common keys.

d1 = {'a': 100, 'b': 200, 'c':300}

d2 = {'a': 300, 'b': 200, 'd':400}

Sample output: Counter ({'a': 400, 'b': 400, 'd': 400, 'c': 300})

13) Write a program that takes sentence as input from the user and computes the frequency of each letter. Use a variable of dictionary type to maintain and show the frequency of each letter.

14) Write a NumPy program to find the most frequent value in an array.

15) Take two NumPy arrays having two dimensions. Concatenate the arrays on axis 1.

16) Write a function that takes two filenames f1 and f2 as input. The function should read the contents of f1 line by line and write them onto f2.

References: -

1) Python Programming- A modular Approach (with Graphics, database, Mobile and Web Applications by Sheetal Taneja and Naveen Kumar, Pearson.

2) Head First Python by Paul Berry, O’Reilly

3) Dive into Python by Mark Pilgrim, APress

4) Beginning Programming with Python Dummies by John Paul Meuller.

5) Programming and Problem Solving Through Python Language, Prof. Satish Jain, Shashi Singh, BPB Publication.

OR

SCRIPTING LANGUAGE LAB(PERL)

Course Code	CSPC307
Course Title	Scripting Language Lab(Perl)
Number of Credits	2 (L: 0, T: 0, P: 4)
Prerequisites	Knowledge of Programming and Linux
Course Category	Programme Core Course

Course Outcomes: -

After the completion of the course Student will be able to

1) The basic programming using Perl scripts **(K3)**.
2) Know about data structures, flow control mechanisms, regular expressions, and subroutines and modules. **(K2)**

Practical Assignments:

1) Write a program that computes the circumference of a circle with a radius of 12.5. The circumference is 2π times the radius (and π approximates 3.141592654).
2) Modify the program from the previous exercise to prompt for, and accept, a radius from the person running the program.
3) Write a program to take in two numbers that prints out the result of the two numbers multiplied together.
4) Write a program that reads in a string and a number, and then prints out the string the number of times requested. (Hint: use the ‘x’ operator)
5) Write a program that reads a list of strings, and prints out the list in reverse order.
6) Write a program that reads in a number and a series of lines, then prints one of the lines from the list, as selected by the number.
7) Write a program that reads in a list of strings, then prints one chosen at random.
8) Write a program that asks for the temperature outside (us oldies work in Farenheit). The program should print **too hot** if the temperature is above 72, and **too cold** otherwise.
9) Write a program that reads in a list of numbers (one per line), until the number 999 is entered, then it prints the sum of all the numbers entered. Be sure not to add the 999.
   For example, if the numbers 1, 2, 3, 999 are entered, the answer is 6 (1+2+3).
10) Write a program that reads in a list of strings (on separate lines), then prints out the list in reverse order. Do this without using the reverse operator.
11) Write a program that reads a series of words (with one word per line) until End-Of File, and then prints a summary of how many times each word was seen.
12) Write a program that acts like ***cat***, but reverses the order of the lines.
13) Construct regular expressions that match:
    a. at least one ‘a’, followed by any number of ‘b’s

b. any number of back-slashes, followed by any number of stars
c. three consecutive copies of whatever is contained within the variable $whatever
d. any five characters, including the newline character
e. the same word written two or more times in a row, where "word" is defined as a non-empty sequence of non-whitespace characters.

14) Write a program that looks through **/home/kiz/test.passwd.file** (on STDIN), printing the login name and real name of each user.
15) Write a program that looks through **/home/kiz/test.passwd.file** (on STDIN), for users with the same first name, and prints out those names.
16) Write a program that accepts a list of words on STDIN and looks for a line containing all five vowels (specifically a,e,i,o,u). Run this this program on */usr/dict/words*. ie, run "*myprog < /usr/dict/words*"

References: -

1) Programming Perl by Larry Wall, Tom Christiansen, & Randal Schwartz; O'REILLY.
2) Learning Perl by Tom Phoenix, Randal L. Schwartz, O'REILLY.
3) Perl Cookbook by Tom Christiansen, Nathan Torkington, O'REILLY.
4) Perl: The Complete Reference by Martin C Brown, McGraw-Hill

****

Operating System Lab

Course Code	CSPC308
Course Title	Operating System Lab
Number of Credits	1 (L: 0, T: 0, P: 2)
Prerequisites	Computer Programming using C
Course Category	Programme Core Course

Course Outcome:-

After completion of the course, students will be able to:

C.O. 1: Learns Operating systems- LINUX/UNIX.**(K2)**
C.O. 2: Understand various UNIX commands on a standard UNIX/LINUX Operating System.**(K2)**
C.O. 3: Apply the scheduling algorithms for the given problem.**(K3)**
C.O. 4: Implement the process synchronous concept using message queue, shared memory . **(K3)**
C.O. 5: Implement the various methods in memory allocation and page replacement algorithm. **(K3)**

List of Practical's/ Activities *(To perform minimum 6 practical)*

1. Revision practice of various commands like man, cp, mv, ln, rm, unlink, mkdir, rmdir, etc and many more that were learnt in IT Workshop course and later.
2. Implement two way process communication using pipes.
3. Implement message queue form of IPC
4. Implement shared memory and semaphore form of IPC
5. Simulate the CPU scheduling algorithms - Round Robin, SJF, FCFS, priority
6. Simulate all FIFO Page Replacement Algorithm using C program
7. Simulate all LRU Page Replacement Algorithms using C program
8. Simulate Paging Technique of Memory Management
9. Practice various commands/utilitiessuch as catnl, uniq, tee, pg, comm, cmp, diff, tr, tar, cpio, mount, umount, find, umask, ulimit, sort, grep, egrep,fgrep cut, paste, join, du, df , ps, who, etc and many more.

Reference Books:

1. Operating System Concepts, Silberschatz, Abraham and Galvin, Peter, Wiley India Limited.
2. UNIX Concepts and Applications, Sumitabha Das, McGraw-Hill Education.
3. Operating System Concepts, Ekta Walia, Khanna Publishing House.

****

Digital Logic Design & Microprocessor Lab

Course Code	CSPC309
Course Title	Digital logic design & Microprocessor Lab
Number of Credits	1 (L: 0, T:0, P: 2)
Prerequisites	--------
Course Category	Programme Core Course

Course Outcomes: -

After Completion of the course students will be able to:

1) Test the digital systems, logic families and logic gates.
2) Construct combinational logical circuit.
3) Construct sequential logical circuit.
4) Use registers and instructions of 8086.
5) Develop assembly language programs using 8086.

Detailed course content:

Sl. No.	Suggested Programme/experiments with Practical outcomes	Linked Unit to the theory paper CSPC305
1	Test truth tables of basic logic gates using Transistor-Transistor Logic (TTL), Integrated Circuits (ICs)	1
2	Check truth tables of universal logic gates (NAND and NOR) using TTL and ICs.	1 *
3	Check De-Morgan’s theorem using ICs	1
4	Convert given expression to Sum of Product (SPO) from using basic logic gates.	2 *
5	Convert given expression to Product of Sum (POS) from using basic logic gates.	2
6	Implement Combinational Circuit using Multiplexer.	2
7	Construct S-R, J-K, D and T flip-flop and verify their truth tables.	3 *
8	Write and execute an Assembly Language Program (ALP) to ass/subtract two 8 bit and 16 bit numbers with the help of programming tools and any simulator.	5 *
9	Write and execute ALP to find sum of series of 8 bit and 16 bit numbers.	5
10	Develop an ALP to multiply two 8 bit and 16 bit numbers. (Unsigned/signed numbers).	5 *
11	Develop an ALP to divide two 8 bit and 16 bit numbers. (Unsigned/signed numbers).	5
12	Write an ALP to add/subtract two BCD numbers.	5
13	Write an ALP to multiply/divide two BCD numbers.	5
14	Develop an ALP to find smallest and largest number from array of n numbers.	5 *
15	Develop an ALP to find largest number from array of n numbers.	5
16	Write an ALP to perform block transfer from one memory location to another.	5 *

(**Note:** The list of experiments is just suggestive. More such practical outcome can be added to attain the Cos and competency. A judicial mix of minimum 12 or more practical need to be performed, out of which the practical marked as ‘*’ are compulsory.

Suggestive list of equipments/instruments required:

Sl. No.	Equipment Name with board specifications
1	Digital Multimeter, pulse generator/functional generation, DC regulated power supply, Bread boards, connecting wires, Stripper, Soldering Gun, Soldering Metal, Flux, IC Tester, LEDs, Digital ICs, Data sheets of ICs used in Lab.
2	Desktop Computer with minimum 2 GB RAM, 500GB HDD, windows 7

	onwards, any editor to write/edit programs, Turbo/Macro Assembler (TASM/MASM), Turbo Linker (TLINK/LINK), Turbo Debugger (TD/Debug), (DOSBOX utility for higher end operating systems), shared Printer. 8086 Microprocessor programming Kit.
3	8086 freeware/open source-based simulator to demonstrate internal functioning of microprocessor.

****

Summer Internship-I

Course Code	CSSI310
Course Title	Summer Internship-I
Number of Credits	2 (L: 0, T: 0, P: 0)
Prerequisites	Nil
Course Category	Internship

Internships may be full-time or part-time; they are full-time in the summer vacation and part-time during the academic session.

Sl. no.	Schedule	Duration	Activities	Credits	Hours of Work
1	Summer Vacation after 2nd Semester	3-4 Weeks	Inter/ Intra Institutional Activities **	2	80 Hours

(** Students are required to be involved in Inter/ Intra Institutional Activities viz; Training with higher Institutions; Soft skill training organized by Training and Placement Cell of the respective Institutions; contribution at incubation/ innovation /entrepreneurship cell of the Institute; participation in conferences/ workshops/ competitions etc.; Learning at Departmental Lab/ Tinkering Lab/ Institutional workshop; Working for consultancy/ research project within the Institutes and Participation in all the activities of Institute’s Innovation Council for e.g.: IPR workshop/Leadership Talks/ Idea/ Design/ Innovation/ Business Completion/ Technical Expos etc.)

Benefits to Students:

1. An opportunity to get hired by the Industry/ organization.
2. Practical experience in an organizational setting.
3. Excellent opportunity to see how the theoretical aspects learned in classes are integrated into the practical world. On-floor experience provides much more professional experience which is often worth more than classroom teaching.
4. Helps them decide if the industry and the profession is the best career option to pursue.
5. Opportunity to learn new skills and supplement knowledge.
6. Opportunity to practice communication and teamwork skills.

7. Opportunity to learn strategies like time management, multi-tasking etc. in an industrial setup.
8. Opportunity to meet new people and learn networking skills.
9. Makes a valuable addition to their resume.
10. Enhances their candidacy for higher education.
11. Creating network and social circle and developing relationships with industry people.
12. Provides opportunity to evaluate the organization before committing to a full-time position.

Course Outcome:-

After completion of the course, students will be able to:

C.O.1: Explain the real life organizational and industrial environment situations (K2).

C.O.2: Develop organizational dynamics in terms of organizational behaviour, culture and professional ethics (K1).

C.O.3: Understand the importance of Team work (K2).

C.O.4: Explain invaluable knowledge and networking experience (K2).

C.O.5: Develop skill to build a relationship with a prospective employer (K3).

Course Content:-

Internships are educational and career development opportunities, providing practical experience in a field or discipline. The Summer Internship-I is a student centric activity that would expose Technical students to the industrial environment, which cannot be simulated in the classroom and hence creating competent professionals for the industry. They are structured, short-term, supervised placements often focused around particular tasks or projects with defined timescales. An internship may be compensated, non-compensated or some time may be paid. The internship has to be meaningful and mutually beneficial to the intern and the organization. It is important that the objectives and the activities of the internship program are clearly defined and understood. Following are the intended objectives of internship training:

1. Will expose Technical students to the industrial environment, which cannot be simulated in the classroom and hence creating competent professionals for the industry.
2. Provide possible opportunities to learn, understand and sharpen the real time technical / managerial skills required at the job.
3. Exposure to the current technological developments relevant to the subject area of training.
4. Experience gained from the ‘Industrial Internship’ in classroom will be used in classroom discussions.

5. Create conditions conducive to quest for knowledge and its applicability on the job.
6. Learn to apply the Technical knowledge in real industrial situations.
7. Gain experience in writing Technical reports/projects.
8. Expose students to the engineer's responsibilities and ethics.
9. Familiarize with various materials, processes, products and their applications along with relevant aspects of quality control.
10. Promote academic, professional and/or personal development.
11. Expose the students to future employers.
12. Understand the social, economic and administrative considerations that influence the working environment of industrial organizations
13. Understand the psychology of the workers and their habits, attitudes and approach to problem solving.

Overall compilation of Internship Activities / Credit Framework:

Major Head of Activity	**Credit**	**Schedule**	**Total Duration**	**Sub Activity Head**	**Proposed Document as Evidence**	**Evaluated by**	**Performance appraisal/ Maximum points/ activity**
Inter/ Intra Institutional Activities	2	Summer Vacation after 2nd Semester	3-4 Weeks	Inter/ Intra Institutional Workshop/ Training	Certificate	Programme Head	Satisfactory/ Good/ Excellent
				Working for consultancy/ research project	Certificate	Programme Head	Satisfactory/ Good/ Excellent
				Festival (Technical / Business / Others) Events	Certificate	Programme Head	Satisfactory/ Good/ Excellent
				Contribution in Incubation/ Innovation/ Entrepreneurship Cell/ Institutional Innovation Council	Certificate	Cell In-charge	Satisfactory/ Good/ Excellent
				Learning at Departmental Lab/Tinkering Lab/ Institutional workshop	Certificate	Cell In-charge	Satisfactory/ Good/ Excellent

STUDENT’S DIARY/ DAILY LOG

The main purpose of writing daily diary is to cultivate the habit of documenting and to encourage the students to search for details. It develops the students’ thought process and reasoning abilities. The students should record in the daily training diary the day-to-day account of the observations, impressions, information gathered and suggestions given, if any. It should contain the sketches & drawings related to the observations made by the students.

The daily training diary should be signed at the end of each day by the supervisor/ in charge of the section where the student has been working. The diary should also be shown to the Faculty Mentor visiting the industry from time to time and get ratified on the day of his visit.

Student’s Diary and Internship Report should be submitted by the students along with attendance record and an evaluation sheet duly signed and stamped by the industry to the Institute immediately after the completion of the training. It will be evaluated on the basis of the following criteria:

a) Regularity in maintenance of the diary.
b) Adequacy & quality of information recorded.
c) Drawings, sketches and data recorded.
d) Thought process and recording techniques used.
e) Organization of the information.

INTERNSHIP REPORT

After completion of Internship, the student should prepare a comprehensive report to indicate what he has observed and learnt in the training period. The student may contact Industrial Supervisor/ Faculty Mentor/TPO for assigning special topics and problems and should prepare the final report on the assigned topics. Daily diary will also help to a great extent in writing the industrial report since much of the information has already been incorporated by the student into the daily diary. The training report should be signed by the Internship Supervisor, TPO and Faculty Mentor. The Internship report will be evaluated on the basis of following criteria:

a) Originality.
b) Adequacy and purposeful write-up.
c) Organization, format, drawings, sketches, style, language etc.
d) Variety and relevance of learning experience.
e) Practical applications, relationships with basic theory and concepts taught in the course.

****