The University of Burdwan

Syllabus of 3 Years Degree / 4 YearsHonours in Computer Science

Under Curriculum and Credit Framework for Undergraduate Programme (CCFUP) in Computer Science as per NEP, 2020. (w.e.f. 2023-2024)

Preamble

The new curriculum of the four-year undergraduate program under NEP, for computer science aims to develop the core competence in computing and problem solving amongst its graduates. Informally, "Learning to learn" has been the motto of the department since its inception. The curriculum thus focuses on building theoretical foundations in computer science to enable its pupils to think critically when challenged with totally different and new problems. It imbibes the following Student-Centric features of NEP 2020:

Flexibility to Exit: In order to support early exits, the curriculum aims to develop employability skills early. As programming is at the heart of computing it is proposed to have two programming courses early so that the students can develop good programming skills in the first year. At the same time students are familiarized with the hardware of computers early on.

Employability: Industry demand in the IT sector has changed considerably in the past few years. With the humongous amount of data coming from all the domains like medical data, social networking data, astronomical data, education, etc., automating information extraction and analysis of data is the only way forward to leverage the available data for the future. This curriculum aims to equip the students with tools and techniques of Artificial Intelligence, Machine Learning and a pathway on Data Science if the student so desires. Having said this, there is no replacement for the foundational courses like programming, data structures and algorithms. With two courses on programming and two courses on data structures and algorithms together, a strong foundation will be laid down for problem solving.

Multidisciplinary/Minor: The curriculum provides two pathways one of computer science (CS) , minor and the other of interdisciplinary, to the students from other disciplines. Those who want to earn a minor in CS will be required to choose the first pathway whereas those who simply want to apply IT in the domain of their interest can choose the second pathway.

Research: With the option to obtain specialization in an area of their choice, the curriculum prepares the students to take up research projects in their final year.

Discipline Specific Graduate Attributes (DSGA)

1. Proficiency in writing readable, correct, efficient, and secure programs of modest complexity.
2. Ability to design efficient algorithms using appropriate data structures for new problems.
3. Understanding of computer architecture, operating systems, computer networks and database managementsystems and their role in the performance of software applications.
4. Understanding of theoretical foundations and limits of computing.
5. Ability to develop good quality software by following the processes of software development life cycle.
6. Ability to extract information and analyze large volumes of data employing a range of techniques forartificial intelligence and learning.
7. Ability to develop an end to end compiler using compiler designing tools and techniques.
8. Ability to protect the data and software from various types of cyber-attacks.

SEMESTER WISE COURSE STRUCTURE

Semester	Course Type & Course Code	Name of the Course	Credit	Lect.	Tuto.	Pract./ Viva	Full Marks	Distribution of Marks		
								Theory	Pract. / Tuto./ Viva-voce	Internal Assessment
I	Major/DS Course (Core) **COMP 1011**	Computer Fundamentals & Digital Logic	4	3	0	1	75	40	20	15
	Minor Course **COMP 1021**	Computer Fundamentals & Digital Logic **(For other allied discipline)**	4	3	0	1	75	40	20	15
	Multi/Interdisciplinary **COMP 1031**	Basic IT tools **(For other discipline)**	3	3	0	0	50	40	NIL	10
	Ability Enhancement Course(AEC)[L_1-1MIL] **… 1041**	Arabic/ Bengali/ Hindi/ Sanskrit/ Santali/ Urdu or Equvlnt. Course from SWAYAM or UGC recognized other platforms	2	2	0	0	50	40	0	10
	Skill Enhancement Course (SEC) **COMP 1051**	Python Programming	3	0	0	3	50	NIL	40	10
	Common Value Added (CVA) Course **CVA 1061**	Environmental Science/ Education	4	3	0	1	100	60	20	20
	Total		**20**				**400**			

Semester	Course Type	Name of the Course	Credit	Lect.	Tuto.	Pract. /Viva	Full Marks	Distribution of Marks		
								Theory	Pract. / Tuto./ Viva-voce	Internal Assessment
II	Major/DS Course (Core) **COMP 2011**	Programming Fundamentals using C	4	3	0	1	75	40	20	15
	Minor Course **COMP 2021**	Python Programming **(For other allied discipline)**	4	3	0	1	75	40	20	15
	Multi/Interdisciplinary **COMP 2031**	Introduction to Internet **(For other discipline)**	3	3	0	0	50	40	NIL	10
	Ability Enhancement Course(AEC)[L_2-1] **ENGL 2041**	Functional English or Equvlnt. Course from SWAYAM or UGC recognized other platforms	2	2	0	0	50	40	0	10
	Skill Enhancement Course (SEC) **COMP 2051**	System Administration & Maintenance	3	0	0	3	50	NIL	40	10
	Common Value Added (CVA) Course **CVA 2061**	Understanding India Digital and Technological Solution Health and Wellness, Yoga Education, Sports and Fitness	4	3/3	1/0	0/1	100	80/60	0/20	20
	Total		**20**				**400**			

Semester	Course Type	Name of the Course	Credit	Lect.	Tuto	Pract/ Viva	Full Marks	Distribution of Marks		
								Theory	Pract/ tuto./ Viva-voce	Internal Assessment
III	Major/DS Course (Core) **COMP 3011**	Discrete Structure	5	5	0	0	75	60	NIL	15
	Major/DS Course (Core) **COMP 3012**	Data Structures	5	4	0	1	75	40	20	15
	Minor Course (Vocational Education. & Training.) **MSR 3021** **OR** **HRM 3021** **OR** **RSA 3021**	(Intermediate Level Course) Medical Sales Representative OR Human Resource Management OR Retail Sales Associate	4	3	1	0	75	60	0	15
	Multi/Interdisciplinary **COMP 3031**	Introduction to Cyber Security (For other discipline)	3	3	0	0	50	40	NIL	10
	Ability Enhancement Course(AEC)[L1-2 MIL] **… 3041**	Arabic/ Bengali/ Hindi/ Sanskrit/ Santali/ Urdu] or Equivalent. Course from SWAYAM or UGC recognized other platform	2	2	0	0	50	40	0	10
	Skill Enhancement Course (SEC) **COMP 3051**	PHP Programming	3	0	0	3	50	NIL	40	10
	Total		**22**				**375**			

Semester	Course Type	Name of the Course	Credit	Lect.	Tuto.	Pract./Viva	Full Marks	Distribution of Marks		
								Theory	Pract. / Tuto./ Viva-voce	Internal Assessment
IV	Major/DS Course (Core) **COMP 4011**	Operating Systems	5	4	0	1	75	40	20	15
	Major/DS Course (Core) **COMP 4012**	Object Oriented Technology using JAVA	5	4	0	1	75	40	20	15
	Major/DS Course (Core) **COMP 4013**	Database Management Systems	5	4	0	1	75	40	20	15
	Minor Course **COMP 4021**	Programming in C (For other allied discipline)	4	3	0	1	75	40	20	15
	Minor Course (other than Computer Science) **… 4021**		4				75			15
	Ability Enhancement Course (AEC)[L_2-2] **ENGL 4041**	Language and creativity or Equvlnt. Course from SWAYAM or UGC recognized other platform	2	2	0	0	50	40	0	10
	Total		**25**				**425**			

Semester	Course Type	Name of the Course	Credit	Lect.	Tuto.	Pract. /Viva	Full Marks	Distribution of Marks		
								Theory	Pract. / Viva- voce	Internal Assessment
V	Major/DS Course (Core) **COMP 5011**	Data Communications & Networking	5	4	0	1	75	40	20	15
	Major/DS Course (Core) **COMP 5012**	Computer Organization	5	4	0	1	75	40	20	15
	Major/DS Course (Core) **COMP 5013**	Software Engineering	5	4	1	0	75	60	0	15
	Minor Course (Vocational Education. & Training.) **MSR 5021** **OR** **HRM 5021** **OR** **RSA 5021**	(Intermediate Level Course) Medical Sales Representative OR Human Resource Management OR Retail Sales Associate	4	3	1	0	75	60	0	15
	Internship (for all students) **INT 5081**		2	0	0	2	50	00-50-00 (Project - 30 + Viva - 20)		
	Total		**21**				**350**			

Semester	Course Type	Name of the Course	Credit	Lect.	Tuto.	Pract./ Viva	Full Marks	Distribution of Marks		
								Theory	Pract. / Viva-voce	Internal Assessment
VI	Major/DS Course (Core) **COMP 6011**	Numerical methods	4	3	0	1	75	40	20	15
	Major/DS Course (Core) **COMP 6012**	Microprocessor	4	3	0	1	75	40	20	15
	Major/DS Course (Core) **COMP 6013**	Theory of Computation	4	3	1	0	75	60	0	15
	Major/DS Course (Core) **COMP 6014**	Artificial Intelligence	4	3	0	1	75	40	20	15
	Minor Course (Vocational Education. & Training.) **MSR 6021** **OR** **HRM 6021** **OR** **RSA 6021**	(Intermediate Level Course) Medical Sales Representative OR Human Resource Management OR Retail Sales Associate	4	3	1	0	75	60	0	15
	Total		**20**				**375**			

Semester-I

Major/DS Course (Core)
COMP 1011: Computer Fundamentals & Digital Logic
Credit: 03 **45 Hours**

Course Objective

This course introduces the students to the fundamental concepts of digital computer organization and design. It aims to develop a basic understanding of the building blocks of the computer system and highlights how these blocks are organized together to architect a digital computer system. The course teaches the fundamentals of digital systems, applying the logic design and development techniques. This course forms the basis for the study of advanced subjects like Computer Architecture and Organization, Microprocessor through Interfacing, VLSI Designing etc.

Course Learning Outcomes

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

i. Acquire the basic knowledge of digital logic to understand digital electronics circuits.

ii. Prepare students to perform the analysis and design of various digital electronic circuits.

iii. Design and simplify combinational and sequential circuits using basic building blocks.

iv. Represent data in binary form, convert numeric data between different number systems and perform arithmetic operations in binary.

v. Simulate the design of a basic computer using a software tool/ digital trainer kit.

Syllabus

Computer Fundamentals (15 Hours)

Introduction to Computer and Problem Solving: Information and Data. Hardware: CPU, Primary and Secondary storage, I/O devices, Bus structure Software: Systems and Application. Generation of Computers: Super, Mainframe, Mini and Personal Computer. Introduction to Programming Languages: Machine Language, Assembly Language, High Level Language. Problem Solving: Flow Charts, Decision Tables and Pseudo codes. Number Systems and Codes: Number representation: Weighted Codes, Non-weighted codes, Positional, Binary, Octal, Hexadecimal, Binary Coded Decimal (BCD), Conversion of bases. Complement notions. Binary Arithmetic, Binary Codes: Gray, Alphanumeric, ASCII, EBCDIC; Single Error-Detecting and Correcting Codes, Hamming Codes, IEEE 754 floating point representation. Boolean algebra: Fundamentals of Boolean algebra, Switches and Inverters, Functionally Complete Gates (AND, OR, NOT), NAND, NOR, Switching function and Boolean Function. De Morgan's Theorem, Minterms, Maxterms, Truth table and minimization of switching function up to four variables, Algebraic and K-map method of Logic circuit synthesis: Two-level and Multi-level

Digital Logic(30 Hours)

Combinational Circuits: Realization of AND , OR Gates using diodes and NOT Gate using transistors, Standard Gate Assemblies, IC chips packaging nomenclature, Half and Full Adder(3 & bit), Multi-bit adders – Ripple carry and Carry Look Ahead Adder, Adder/subtractor, BCD-Adder, Data selectors/multiplexers – expansions, reductions, function realization, universal function 5 realization, multi-function realization, Decoders: function realization, De-multiplexer and function realization, Encoder, Priority Encoder, Parity bit Generator/checker, Gray Code Generator, Code Converters, Keyboard encoder, Seven segment display unit, Comparators. Sequential Circuits: Model of Sequential computing, Difference between Combinational and Sequential circuit, RS-Latch: using NAND and NOR Gates, RS Latch as a Static RAM Cell, Problems of Basic Latch circuits, Digital Clock – Duty Cycle, Rising time, Falling time, Clocked Flip Flops - SR, JK, D, T, Level Trigger and Edge Trigger, Excitation Functions of each flip-flops, Flip-flops with Preset and Clear, Application of Flip-flops: Asynchronous Counter(UP/DOWN) up to 4-bit counter, Decade Counter, Mod – n Counter, Finite State machine Model – State Transition Diagram and Table, Synchronous Counters – different mod counters, Ring counter, Johnson's Counter, Registers, Registers with parallel load, Shift Registers.

Practical :: Digital Circuit Design

Credit: 01 **30 Hours**

Combinational Circuits:

1) Implement Half Adder/Half Subtractor/Full Adder/Full Subtractor using Logic Gates. Realize a logic function using basic/universal gates in SOP and POS form. Study the functionalities of 7483 and design a BCD adder using 7483 or equivalent.
2) Design of two level AND – OR, NAND –NAND, NOR-NOR circuits to realize any truth table. Realize XOR in two level and multilevel.
3) Design a 4 bit 2's complement adder – subtractor unit using 7483 or equivalent and XOR gates.
4) Design a circuit to convert BCD numbers to corresponding gray codes.
5) Design a 4:1 MUX using NAND gates. Study of 74153 and 74151. Design Full Adder/Subtractor using MUX.
6) Design a 2:4 decoder using NAND gates. Study of 74155 and 74138. Design Full Adder/Subtractor using decoders.
7) Design a parity generator/checker using basic gates.
8) Design magnitude comparator using basic/universal gates. Study of 7485.
9) Design a seven-segment display unit.

Sequential Circuits:

1) Realize S-R, D, J-K and T flip-flop using basic gates. (Study the undefined state in S-R flip-flop).
2) Design a shift register (shift left and shift right) using flip-flops. (Study the functional characteristic of IC 74194 with emphasis on timing diagram).
3) Design Asynchronous and Synchronous counters. Study of IC 74193.
4) Study the functional characteristics of RAM IC chip. Study of open collector and tri-state output. Horizontal and vertical expansion of RAM chips by cascading. Use 74189, 7489, 2114 or any available chip.

Reference Books:

1. Digital Logic and Computer Design by M.Morris Mano, PHI
2. Digital Fundamentals by Floyd, Pearson Education
3. Digital Principle and Applications by Malvino & Leach, TMH
4 P. K. Sinha & Priti Sinha , Computer Fundamentals‖, BPB Publications, 2007.
5 Dr. Anita Goel, Computer Fundamentals, Pearson Education, 2010.

Skill Enhancement Course (SEC)

Credit: 03 **90 Hours**

COMP 1051: Programming in Python (Practical)

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

Planning the Computer Program: Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation. (20 Hrs)

Techniques of Problem Solving: Flowcharting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming.(20 Hrs)

Overview of Programming: Structure of a Python Program, Elements of Python (10 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.(20 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, function , structures (20Hrs)

Reference Books

1. T. Budd, Exploring Python, TMH, 1st Ed, 2011
2. Python Tutorial/Documentation www.python.or 2015
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/

Lab Work:

Section: A (Simple programs)

- Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon user's 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 <80Grade C:

  Percentage>=60 and <70Grade 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
   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. velocity wrt time (v=u+at)
   II. distance wrt time (s=u*t+0.5*a*t*t)
   III. distance wrt velocity (s=(v*v-u*u)/2*a)

Minor Courses (For other allied discipline):

Semester-I

COMP 1021: Computer Fundamentals & Digital Logic

Credit: 03 **45 Hours**

Course Objective

This course introduces the students to the fundamental concepts of digital computer organization and design. It aims to develop a basic understanding of the building blocks of the computer system and highlights how these blocks are organized together to architect a digital computer system. The course teaches the fundamentals of digital systems, applying the logic design and development techniques. This course forms the basis for the study of advanced subjects like Computer Architecture and Organization, Microprocessor through Interfacing, VLSI Designing etc.

Course Learning Outcomes

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

i. Acquire the basic knowledge of digital logic to understand digital electronics circuits.

ii. Prepare students to perform the analysis and design of various digital electronic circuits.

iii. Design and simplify combinational and sequential circuits using basic building blocks.

iv. Represent data in binary form, convert numeric data between different number systems and perform arithmetic operations in binary.

v. Simulate the design of a basic computer using a software tool/ digital trainer kit.

Syllabus

Computer Fundamentals (15 Hours)

Introduction to Computer and Problem Solving: Information and Data. Hardware: CPU, Primary and Secondary storage, I/O devices, Bus structure Software: Systems and Application. Generation of Computers: Super, Mainframe, Mini and Personal Computer. Introduction to Programming Languages: Machine Language, Assembly Language, High Level Language. Problem Solving: Flow Charts, Decision Tables and Pseudo codes. Number Systems and Codes: Number representation: Weighted Codes, Non-weighted codes, Positional, Binary, Octal, Hexadecimal, Binary Coded Decimal (BCD), Conversion of bases. Complement notions. Binary Arithmetic, Binary Codes: Gray, Alphanumeric, ASCII, EBCDIC; Single Error-Detecting and Correcting Codes, Hamming Codes, IEEE 754 floating point representation. Boolean algebra: Fundamentals of Boolean algebra, Switches and Inverters, Functionally Complete Gates (AND, OR, NOT), NAND, NOR, Switching function and Boolean Function. De Morgan's Theorem, Minterms, Maxterms, Truth table and minimization of switching function up to four variables, Algebraic and K-map method of Logic circuit synthesis: Two-level and Multi-level.

Digital Logic(30 Hours)

Combinational Circuits: Realization of AND ,OR Gates using diodes and NOT Gate using transistors, Standard Gate Assemblies, IC chips packaging nomenclature, Half and Full Adder(3 & bit), Multi-bit adders – Ripple carry and Carry Look Ahead Adder, Adder/subtractor, BCD-Adder, Data selectors/multiplexers – expansions, reductions, function realization, universal function 5 realization, multi-function realization, Decoders: function realization, De-multiplexer and function realization, Encoder, Priority Encoder, Parity bit Generator/checker, Gray Code Generator, Code Converters, Keyboard encoder, Seven segment display unit, Comparators. Sequential Circuits: Model of Sequential computing, Difference between Combinational and Sequential circuit, RS-Latch: using NAND and NOR Gates, RS Latch as a Static RAM Cell, Problems of Basic Latch circuits, Digital Clock – Duty Cycle, Rising time, Falling time, Clocked Flip Flops - SR, JK, D, T, Level Trigger and Edge Trigger, Excitation Functions of each flip-flops, Flip-flops with Preset and Clear, Application of Flip-flops: Asynchronous Counter(UP/DOWN) up to 4-bit counter, Decade Counter, Mod – n Counter, Finite State machine Model – State Transition Diagram and Table, Synchronous Counters – different mod counters, Ring counter, Johnson's Counter, Registers, Registers with parallel load, Shift Registers.

Practical :: Digital Circuit Design

Credit: 01 **30 Hours**

Combinational Circuits:

1) Implement Half Adder/Half Subtractor/Full Adder/Full Subtractor using Logic Gates. Realize a logic function using basic/universal gates in SOP and POS form. Study the functionalities of 7483 and design a BCD adder using 7483 or equivalent.
2) Design of two level AND – OR, NAND –NAND, NOR-NOR circuits to realize any truth table. Realize XOR in two level and multilevel.
3) Design a 4 bit 2's complement adder – subtractor unit using 7483 or equivalent and XOR gates.
4) Design a circuit to convert BCD numbers to corresponding gray codes.
5) Design a 4:1 MUX using NAND gates. Study of 74153 and 74151. Design Full Adder/Subtractor using MUX.
6) Design a 2:4 decoder using NAND gates. Study of 74155 and 74138. Design Full Adder/Subtractor using decoders.
7) Design a parity generator/checker using basic gates.
8) Design magnitude comparator using basic/universal gates. Study of 7485.
9) Design a seven-segment display unit.

Sequential Circuits:

i. Realize S-R, D, J-K and T flip-flop using basic gates. (Study the undefined state in S-R flip-flop).
ii. Design a shift register (shift left and shift right) using flip-flops. (Study the functional characteristic of IC 74194 with emphasis on timing diagram).
iii. Design Asynchronous and Synchronous counters. Study of IC 74193.
iv. Study the functional characteristics of RAM IC chip. Study of open collector and tri-state output. Horizontal and vertical expansion of RAM chips by cascading. Use 74189, 7489, 2114 or any available chip.

Reference Books

a. Digital Logic and Computer Design by M.Morris Mano, PHI
b. Digital Fundamentals by Floyd, Pearson Education
c. Digital Principle and Applications by Malvino & Leach, TMH
d. P. K. Sinha & Priti Sinha , Computer Fundamentals‖, BPB Publications, 2007.
e. Dr. Anita Goel, Computer Fundamentals, Pearson Education, 2010.

Multi/Interdisciplinary courses

(For other discipline)

Credit: 03 **45 Hours**

COMP 1031: Basic IT Tools(Theory)

Course Objective

The goal of this course is to present overview of IT tools used in day to day use of computers and data base operations. The Course has been designed to provide knowledge on various hardware and software components of computer, operating system, various packages used for different applications, data base concepts & operations and various issues related to IT and application of IT.

Course Learning Outcomes:

On successful completion of the Course, a student will :

i. Acquire the foundation level knowledge required to understand computer and its operations.
ii. Understand the hardware and software components of the computer.
iii. Understand the basic concept of operating system and get knowledge about various different operating systems.
iv. Understand to use the packages of word processing, spread sheet and presentation in detail.
v. Understand various data base concepts and operations.
vi. Understand the issues related to IT and IT applications
vii. Prepare research and academic related presentations.

Syllabus

Introduction – Introduction to computers – Evolution – Generation of Computers – Computers Hierarchy – Applications of Computers. (5 Hrs)

Windows Basics – Introduction to word – Editing a document - Move and Copy text - Formatting text & Paragraph – Enhancing document – Columns, Tables and Other features.(10 Hrs)

Introduction to worksheet and shell – getting started with Excel – Editing cell & using Commands and functions – Moving & Copying , Inserting & Deleting Rows & Columns - Printing work sheet.(5 Hrs)

Creating charts – Naming ranges and using statistical, math and financial functions, database in a worksheet – Additional formatting commands and drawing toolbar – other commands & functions – multiple worksheet and macros.(10 Hrs)

Introduction to Database Development: Database Terminology, Objects, Creating Tables, working with fields, understanding Data types, Changing table design, Assigning Field Properties, Setting Primary Keys, Select data with queries: Creating simple Query by design & by wizard (10 Hrs)

Overview of Power point – presenting shows for corporate and commercial using Power point –Introduction to Desktop publishing – Computer viruses – Introduction to Internet – Web features.(5 hrs)

Reference Books:

i. Swinford, E., Dodge, M., Couch, A., Melton, B. A. (2013). Microsoft OfficeProfessional 2013. United States: O'Reilly Media.
ii. Wang, W. (2018). Office 2019 For Dummies. United States: Wiley. Microsoft Lambert, J. (2019). Microsoft Word 2019 Step by Step. United States: Pearson Education.
iii. Jelen, B. (2013). Excel 2013 Charts and Graphs. United Kingdom: Que.
iv. Alexander, M., Jelen, B. (2013). Excel 2013 Pivot Table Data Crunching. UnitedKingdom: Pearson Education.
v. Alexander, M., Kusleika, R. (2018). Access 2019 Bible. United Kingdom: Wiley.

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+……
4. WAP to compute the sum of the first n terms of the following series S =1-2+3-4+5…….
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-valuedelements
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! --------- n/n!

27. WAP to calculate the sum and product of two compatible matrices.

Reference Books

1. C Programming by Karnighan, & Ritchie, PHI
2. Programming through C by Richard Johnsonbaugh and Martin Kalin, Pearson Education
3. Programming in Ansi 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,
- □ Kernal 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 <80Grade C:

  Percentage>=60 and <70Grade 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>=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**
  - The Head, the Body
  - Colors, Attributes
  - Lists, ordered and unordered
- **Unit-III: Links**
  - Introduction
  - Relative Links, Absolute Links
  - Link Attributes

- Using the ID Attribute to Link Within a Document

- **Unit-IV: Images**
  - Putting an Image on a Page
  - Using Images as Links
  - Putting an Image in the Background

- **Unit V: – Tables**
  - Creating a Table
  - Table Headers
  - Captions
  - Spanning Multiple Columns
  - Styling Table

Reference Books

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

Semester-III

Major/DS Course (Core)

COMP 3011: Discrete Structures (Theory)

Credit: 5 **75 hour**

Course Objective

This course is designed as a foundational course to make students learn about the mathematical constructs that are used in Computer Science such as Boolean algebra, sets, relations, functions, principles of counting, and recurrences. In this course, the knowledge of mathematical notation, ideas and concepts learnt at the pre-college levels is extended to orient the students towards mathematical thinking required in Computer Science.

Course Learning Outcomes

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

1. Relate mathematical concepts and terminology to examples in the domain of Computer Science.
2. Model real world problems using various mathematical constructs.
3. Use different proofing techniques; construct simple mathematical proofs using logical arguments.
4. Formulate mathematical claims and construct counter examples

Syllabus

1. Introduction: Sets - finite and Infinite sets, uncountably Infinite Sets; functions, relations, Properties of Binary Relations, Closure, Partial Ordering Relations; counting - Pigeonhole Principle, Permutation and Combination; Mathematical Induction, Principle of Inclusion and Exclusion. (10 Hrs)
2. Growth of Functions: Asymptotic Notations, Summation formulas and properties, Bounding Summations, approximation by Integrals. (10 Hrs)
3. Recurrences: Recurrence Relations, Generating functions, Linear Recurrence Relations with constant coefficients and their solution, Substitution Method, Recurrence Trees, Master Theorem. (15 Hrs)
4. Graph Theory: Basic Terminology, Models and Types, multi-graphs and weighted graphs, Graph Representation, Graph Isomorphism, Connectivity, Euler and Hamiltonian Paths and Circuits, Planar Graphs, Graph Coloring, Trees, Basic Terminology and properties of Trees, Introduction to Spanning Trees. (20 Hrs)
5. Propositional Logic: Logical Connectives, Well-formed Formulas, Tautologies, Equivalences, Inference Theory. (20 Hrs)

Recommended Books:

1. C.L. Liu , D.P. Mahopatra, Elements of Discrete mathematics, 2nd Edition , Tata McGraw Hill, 1985,
2. Kenneth Rosen, Discrete Mathematics and Its Applications, Sixth Edition ,McGraw Hill 2006
3. T.H. Coremen, C.E. Leiserson, R. L. Rivest, Introduction to algorithms, 3rd edition Prentice Hall on India, 2009

4. M. O. Albertson and J. P. Hutchinson, Discrete Mathematics with Algorithms , John wiley Publication, 1988
5. J. L. Hein, Discrete Structures, Logic, and Computability, 3rd Edition, Jones and Bartlett Publishers, 2009 6. D.J. Hunter, Essentials of Discrete Mathematics

Major/DS Course (Core): Data Structures

COMP 3012 : Data Structures (Theory)

Credit: 4 60 Hrs

Course Objective

The course aims at developing the ability to use basic data structures like arrays, stacks, queues, lists, trees to solve problems. C++ is chosen as the language to understand implementation of these data structure.

Course Learning Outcomes

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

1. Compare two functions for their rates of growth.
2. Understand abstract specification of data-structures and their implementation.
3. Compute time and space complexity of operations on a data-structure.
4. Identify the appropriate data structure(s) for a given application and understand the trade-offs involved in terms of time and space complexity.
5. Apply recursive techniques to solve problems

Syllabus

1. Arrays (5 Hrs)

Single and Multi-dimensional Arrays, Sparse Matrices (Array and Linked Representation)

2. Stacks (5 Hrs)

Implementing single / multiple stack/s in an Array; Prefix, Infix and Postfix expressions, Utility and conversion of these expressions from one to another; Applications of stack; Limitations of Array representation of stack

3. Linked Lists (10 Hrs)

Singly, Doubly and Circular Lists (Array and Linked representation); Normal and Circular representation of Stack in Lists; Self Organizing Lists; Skip Lists

4. Queues (5 Hrs)

Array and Linked representation of Queue, De-queue, Priority Queues,
Queue using Stack

5. Recursion (5 Hrs)

Developing Recursive Definition of Simple Problems and their implementation; Advantages

and Limitations of Recursion; Understanding what goes behind Recursion (Internal Stack Implementation)

6. Trees (20 Hrs)

Introduction to Tree as a data structure; Binary Trees (Insertion, Deletion , Recursive and Iterative Traversals on Binary Search Trees); Threaded Binary Trees (Insertion, Deletion, Traversals);

7. Searching and Sorting (5 Hrs)

Linear Search, Binary Search, Comparison of Linear and Binary Search, Selection Sort, Bubble Sort, Insertion Sort, Shell Sort, Comparison of Sorting Techniques

8. Hashing (5 Hrs)

Introduction to Hashing, Deleting from Hash Table, Efficiency of Rehash Methods, Hash Table Reordering, Resolving collusion by Open Addressing, Coalesced Hashing, Separate Chaining, Dynamic and Extendible Hashing, Choosing a Hash Function, Perfect Hashing Function

Reference Books

1. Adam Drozdek, "Data Structures and algorithm in C++", Third Edition, Cengage Learning, 2012.

2. SartajSahni, Data Structures, "Algorithms and applications in C++", Second Edition, Universities Press, 2011.

3. Aaron M. Tenenbaum, Moshe J. Augenstein, YedidyahLangsam, "Data Structures Using C and C++:, Second edition,

PHI, 2009.

4. Robert L. Kruse, "Data Structures and Program Design in C++", Pearson,1999.

5. D.S Malik, Data Structure using C++,Second edition, Cengage Learning, 2010

6. Mark Allen Weiss, "Data Structures and Algorithms Analysis in Java", Pearson Education, 3rd edition, 2011

7. Aaron M. Tenenbaum, Moshe J. Augenstein, YedidyahLangsam, "Data Structures Using Java, 2003.

8. Robert Lafore, "Data Structures and Algorithms in Java, 2/E", Pearson/ Macmillan Computer Pub,2003

9. John Hubbard, "Data Structures with JAVA", McGraw Hill Education (India) Private Limited; edition, 2009

10. Goodrich, M. and Tamassia, R. "Data Structures and Algorithms Analysis in Java", 4th Edition, Wiley,2013

11. Herbert Schildt, "Java The Complete Reference (English) 9th Edition Paperback", Tata McGraw Hill, 2014.

12. D. S. Malik, P.S. Nair, "Data Structures Using Java", Course Technology, 2003.

Data Structures Lab

Credit: 1 30 Hrs

All programs should be developed in C/Python

1. Write a program to search an element from a list. Give user the option to perform Linear or Binary search. Use Template functions.
2. WAP using templates to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort or Selection sort.
3. Implement Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list and concatenate two linked lists (include a function and also overload operator +).
4. Implement Doubly Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
5. Implement Circular Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
6. Perform Stack operations using Linked List implementation.
7. Perform Stack operations using Array implementation. Use Templates.
8. Perform Queues operations using Circular Array implementation. Use Templates.
9. Create and perform different operations on Double-ended Queues using Linked List implementation.
10. WAP to scan a polynomial using linked list and add two polynomial.
11. WAP to calculate factorial and to compute the factors of a given no. (i)using recursion, (ii) using iteration
12. (ii) WAP to display fibonacci series (i)using recursion, (ii) using iteration
13. WAP to calculate GCD of 2 number (i) with recursion (ii) without recursion
14. WAP to create a Binary Search Tree and include following operations in tree:

(a) Insertion (Recursive and Iterative Implementation)

(b) Deletion by copying

(c) Deletion by Merging

(d) Search a no. in BST

(e) Display its preorder, postorder and inorder traversals Recursively

(f) Display its preorder, postorder and inorder traversals Iteratively

(g) Display its level-by-level traversals

(h) Count the non-leaf nodes and leaf nodes

(i) Display height of tree

(j) Create a mirror image of tree

(k) Check whether two BSTs are equal or not

15. WAP to convert the Sparse Matrix into non-zero form and vice-versa.

16. WAP to reverse the order of the elements in the stack using additional stack.
17. WAP to reverse the order of the elements in the stack using additional Queue.
18. WAP to implement Diagonal Matrix using one-dimensional array.
19. WAP to implement Lower Triangular Matrix using one-dimensional array.
20. WAP to implement Upper Triangular Matrix using one-dimensional array.
21. WAP to implement Symmetric Matrix using one-dimensional array.
22. WAP to create a Threaded Binary Tree as per inorder traversal, and implement operations like finding the successor / predecessor of an element, insert an element, inorder traversal.
23. WAP to implement various operations on AVL

Skill Enhancement Course (SEC)

COMP 3051: PHP Programming (Practical)

Credit :3 90Hrs

Course Objective

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

Course learning outcome

In this course, you will learn:

- How to use PHP's built-in server to serve static resources
- How to use PHP to add some dynamic aspects to our pages
- How to use HTML forms
- The difference between GET and POST requests
- How to use cookies to store some data in the browser and pass it to the next request
- How to use a session cookie to store data on the server instead of in the browser
- How to build an authentication system
- How to restructure the project
- How to upload files to the website
- How to build a custom solution that catches PHP errors and exceptions and shows a proper error page for them
- How to describe and test all the features you've built in this course using an automated test runner

Syllabus

Introduction to PHP: 90 hrs

- PHP introduction, inventions and versions, important tools and software requirements (like Web Server, Database,Editors etc.)
  PHP with other technologies,
  Scope of PHP
- Basic Syntax, PHP variables and constants
- Types of data in PHP , Expressions, scopes of a variable (local, global)
- PHP Operators: Arithmetic, Assignment, Relational, Logical operators, Bitwise, ternary and MOD operator.
- PHP operator Precedence and associativity

Handling HTML form with PHP:
Capturing Form Data

- GET and POST form methods
- Dealing with multi value fields
- Redirecting a form after submission

PHP conditional events and Loops:
PHP IF Else conditional statements (Nested IF and Else)

- Switch case, while ,For and Do While Loop
- Goto , Break ,Continue and exit

PHP Functions:
Function, Need of Function , declaration and calling of a function

- PHP Function with arguments, Default Arguments in Function
- Function argument with call by value, call by reference
- Scope of Function Global and Local

String Manipulation and Regular Expression:

- Creating and accessing String , Searching & Replacing String
- Formatting, joining and splitting String , String Related Library functions
- Use and advantage of regular expression over inbuilt function
- Use of preg_match(), preg_replace(), preg_split() functions in regular expression

Array:

- Anatomy of an Array ,Creating index based and Associative array ,Accessing array
- Looping with Index based array, with associative array using each() and foreach()
- Some useful Library function

Reference Books:

1. Steven Holzner, "PHP: The Complete Reference Paperback", McGraw Hill Education (India), 2007.
2. Timothy Boronczyk, Martin E. Psinas, "PHP and MYSQL (Create-Modify-Reuse)", WileyIndia Private Limited, 2008.
3. Robin Nixon, "Learning PHP, MySQL, JavaScript, CSS & HTML5", 3rd Edition Paperback, O'reilly, 2014.
4. Luke Welling, Laura Thompson, PHP and MySQL Web Development", 4th Edition, Addition Paperback, Addison-Wesley Professsional,2008.
5. David Sklar, Adam Trachtenberg, "PHP Cookbook: Solutions & Examples forPHP Programmers", 2014.

Lab Work:

1. Create a PHP page using functions for comparing three integers and print the largest number.
2. Write a function to calculate the factorial of a number (non-negative integer). The functionaccept the number as an argument.
3. WAP to check whether the given number is prime or not.
4. Create a PHP page which accepts string from user. After submission that page displays thereverse of provided string.
5. Write a PHP function that checks if a string is all lower case.
6. Write a PHP script that checks whether a passed string is palindrome or not? (A palindromeis word, phrase, or sequence that reads the same backward as forward, e.g., madam ornurses run)
7. WAP to sort an array.
8. Write a PHP script that removes the whitespaces from a string.

   Sample string : 'The quick " "
   brown fox'
   Expected Output :
   Thequick""brownfox
9. Write a PHP script that finds out the sum of first n odd numbers.
10. Create a login page having user name and password. On clicking submit, a welcome message should be displayed if the user is already registered (i.e.name is present in the database) otherwise error message should be displayed.
11. Write a PHP script that checks if a string contains another string.
12. Create a simple 'birthday countdown' script, the script will count the number of days between current day and birth day.
13. Create a script to construct the following pattern, using nested for loop.

    *
    * *
    * * *
    * * * *
    * * * * *
14. Write a simple PHP program to check that emails are valid.
15. WAP to print first n even numbers.
16. $color = array('white', 'green', 'red")

    Write a PHP script which will display the colors in the followingway : Output : white, green, red,

    • green
    • red
    • white
17. Using switch case and dropdown list display a ―Hello‖ message depending on the language selected in drop down list.
18. Write a PHP program to print Fibonacci series using recursion.
19. Write a PHP script to replace the first 'the' of the following string with 'That'.

Sample : 'the quick brown fox jumps over the lazy dog.'
Expected Result : The quick brown fox jumps over the lazy dog.

Multi/Interdisciplinary Courses (For Other discipline)

COMP 3031: Introduction to Cyber Security (Theory)

Credit: 3 45 Hrs

Course Objective

It aims to educate individuals about potential cyber threats, best practices for safeguarding sensitive information, and how to respond effectively in case of a security incident.

Course learning outcome

Upon completion of the degree program, students will be able to:

1. Analyse and evaluate the cyber security needs of an organization.
2. Conduct a cyber security risk assessment.
3. Measure the performance and troubleshoot cyber security systems.
4. Implement cyber security solutions.
5. Be able to use cyber security, information assurance, and cyber/computer forensics software/tools.
6. Identify the key cyber security vendors in the marketplace.
7. Design and develop a security architecture for an organization.

Syllabus

Introduction to Cyber Space

History of Internet, Cyber Crime , Information Security ,Computer Ethics and Security , Choosing the Best Browser according to the requirement and email security, Guidelines to choose web browsers , Securing web browser , Antivirus , Email security 5 hrs

Guidelines for secure password and wi-fi security 5 hrs

Guidelines for setting up a Secure password , Two-steps authentication ,Password Manager ,Wi-Fi Security

Guidelines for social media and basic Windows security

Guidelines for social media security , Tips and best practices for safer Social Networking , Basic Security for Windows ,User Account Password 5 hrs

Smartphone security guidelines 5 hrs

Introduction to mobile phones , Smartphone Security ,Android Security ,IOS Security

Cyber Security Initiatives in India 5 hrs

Counter Cyber Security Initiatives in India , Cyber Security Exercise ,Cyber Security Incident Handling

Cyber Security Assurance

Online Banking, Credit Card and UPI Security 5 hrs

Online Banking Security , Mobile Banking Security , Security of Debit and Credit Card , UPI Security

Micro ATM, e-wallet and POS Security

Security of Micro ATMs , e-wallet Security Guidelines ,Security Guidelines for Point of Sales(POS)

Social Engineering 5 hrs

Social Engineering , Types of Social Engineering ,How Cyber Criminal Works ,How to prevent for being a victim of Cyber Crime

Cyber Security Threat Landscape and Techniques 5 hrs

Cyber Security Threat Landscape , Emerging Cyber Security Threats , Cyber Security Techniques ,Firewall

IT Security Act and Misc. Topics

IT Act , Hackers-Attacker-Countermeasures ,Web Application Security ,Digital Infrastructure Security ,Defensive Programming 5 hrs

Information Destroying and Recovery Tools

Recovering from Information Loss , Destroying Sensitive Information , CCleaner for Windows

Reference Books:

i) Cybersecurity: The Beginner's Guide: A comprehensive guide to getting started in cybersecurity, Dr. Erdal Ozkaya

ii) Introduction to Cyber Security: concepts, principles, technologies and practices ,Ajay Singh

Semester -IV

Major/DS Course (Core):
COMP 4011: Operating Systems (Theory)

Credit: 4 **60 Hrs**

Course Objective

The course provides concepts that underlie all operating systems not tied to any particular operating system. The emphasis is done to explain the need and structure of an operating system using its common services such as process management (creation, termination etc.), CPU Scheduling, Process Synchronization, Handling Deadlocks, main memory management, virtual Draft memory, secondary memory management. The course also introduces various scheduling algorithms and structures/techniques used by operating systems to provide these services.

Course Learning Outcomes

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

1. Understand the need of an Operating System & Define Multiprogramming and Multithreading concepts.
2. Implement Process Synchronization service (Critical Section, Semaphores), CPU scheduling service with various algorithms.
3. Learn Main memory Management (Paging, Segmentation) algorithms, Handling of Deadlocks
4. Identify and appreciate the File systems Services, Disk Scheduling service

Syllabus

1. Introduction (10 Hrs)

Basic OS functions, resource abstraction, types of operating systems–multiprogramming systems, batch systems , time sharing systems; operating systems for personal computers & workstations, process control & real time systems.

2. Operating System Organization (6 Hrs)

Processor and user modes, kernels, system calls and system programs.

3. Process Management (20 Hrs)

System view of the process and resources, process abstraction, process hierarchy, threads, threading issues, thread libraries; Process Scheduling, non-pre-emptive and pre-emptive scheduling algorithms; concurrent and processes, critical section, semaphores, methods for inter-process communication; deadlocks.

4. Memory Management (10 Hrs)

Physical and virtual address space; memory allocation strategies –fixed and variable partitions, paging, segmentation, virtual memory

5. File and I/O Management (10 Hrs)

Directory structure, file operations, file allocation methods, device management.
6. Protection and Security (4 Hrs)
Policy mechanism, Authentication, Internal access Authorization.

Recommended Books:

1. A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8th Edition, John Wiley Publications 2008.
2. A.S. Tanenbaum, Modern Operating Systems, 3rd Edition, Pearson Education 2007.
3. G. Nutt, Operating Systems: A Modern Perspective, 2nd Edition Pearson Education 1997.
4. W. Stallings, Operating Systems, Internals & Design Principles , 5th Edition, Prentice Hall of India. 2008.
5. M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992.

Operating Systems Lab

Credit: 1 30Hrs

All programs should be developed in C/Java/Python

1. Write a program (using fork() and/or exec() commands) where parent and child execute:
A) same program, same code. B) same program, different code.
C) before terminating, the parent waits for the child to finish its task.
2. Write a program to report behaviour of Linux kernel including kernel version, CPU type and model. (CPU information)
3. Write a program to report behaviour of Linux kernel including information on configured memory, amount of free and used memory. (memory information)
4. Write a program to print file details including owner access permissions, file access time, where file name is given as argument.
5. Write a program to copy files using system calls.
6. Write program to implement FCFS scheduling algorithm.
7. Write program to implement Round Robin scheduling algorithm.
8. Write program to implement SJF scheduling algorithm.
9. Write program to implement non-preemptive priority based scheduling algorithm.
10. Write program to implement preemptive priority based scheduling algorithm.
11. Write program to implement SRJF scheduling algorithm.
12. Write program to calculate sum of n numbers using thread library.
13. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.

Major/DS Course (Core):

COMP 4012: Programming in Java (Theory)

Credit: 4 **60 Hrs**

Course Objective

This course is designed to develop structured as well as object-oriented programming skills using Java programming language. The course provides a complete understanding of the object-oriented programming features, namely Encapsulation, Abstraction, Inheritance and Polymorphism along with an in-depth knowledge of Java constructs.

Course Learning Outcomes

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

1. Explain significance of object-oriented paradigm.
2. Solve programming problems using Java.
3. Create classes and reuse them.
4. Implement programs using dynamic memory allocation.
5. Handle external files as well as exceptions.

Syllabus

1. Introduction to Java (4 Hrs)

Java Architecture and Features, Understanding the semantic and syntax differences between C++ and Java, Compiling and Executing a Java Program, Variables, Constants, Keywords Data Types, Operators (Arithmetic, Logical and Bitwise) and Expressions, Comments, Doing Basic Program Output, Decision Making Constructs (conditional statements and loops) and Nesting, Java Methods (Defining, Scope, Passing and Returning Arguments, Type Conversion and Type and Checking, Built-in Java Class Methods),

2. Arrays, Strings and I/O (8 Hrs)

Creating & Using Arrays (One Dimension and Multi-dimensional), Referencing Arrays Dynamically, Java Strings: The Java String class, Creating & Using String Objects, Manipulating Strings, String Immutability & Equality, Passing Strings To & From Methods,

String Buffer Classes. Simple I/O using System.out and the Scanner class, Byte and Character streams, Reading/Writing from console and files.

3. Object-Oriented Programming Overview (14 Hrs)

Principles of Object-Oriented Programming, Defining & Using Classes, Controlling Access to Class Members, Class Constructors, Method Overloading, Class Variables & Methods, Objects as parameters, final classes, Object class, Garbage Collection.

4. Inheritance, Interfaces, Packages, Enumerations, Autoboxing and Metadata (14 Hrs)

Inheritance: (Single Level and Multilevel, Method Overriding, Dynamic Method Dispatch, Abstract Classes), Interfaces and Packages, Extending interfaces and packages, Package and Class Visibility, Using Standard Java Packages (util, lang, io, net), Wrapper Classes, Autoboxing/Unboxing, Enumerations and Metadata.

5. Exception Handling, Threading, Networking and Database Connectivity (10 Hrs)

Exception types, uncaught exceptions, throw, built-in exceptions, Creating your own exceptions; Multi-threading: The Thread class and Runnable interface, creating single and multiple threads, Thread

prioritization, synchronization and communication, suspending/resuming threads.
6. Applets and Event Handling (10 Hrs)
Java Applets: Introduction to Applets, Writing Java Applets, Working with Graphics, Incorporating Images & Sounds. Event Handling Mechanisms, Listener Interfaces, Adapter and Inner Classes. The design and Implementation of GUIs using the AWT controls.

Programming in Java(Practical)

Credit: 1 **30 Hrs**

1. To find the sum of any number of integers entered as command line arguments
2. To find the factorial of a given number
3. To learn use of single dimensional array by defining the array dynamically.
4. To learn use of .lenth in case of a two dimensional array
5. To convert a decimal to binary number
6. To check if a number is prime or not, by taking the number as input from the keyboard
7. To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument
8. Write a program that show working of different functions of String and StringBufferclasss like setCharAt(, setLength(), append(), insert(), concat()and equals().
9. Write a program to create a ―distance‖ class with methods where distance is computed in terms of feet and inches, how to create objects of a class and to see the use of this pointer
10. Modify the ―distance‖ class by creating constructor for assigning values (feet and inches) to the distance object. Create another object and assign second object as reference variable to another object reference variable. Further create a third object which is a clone of the first object.
11. Write a program to show that during function overloading, if no matching argument is found, then java will apply automatic type conversions(from lower to higher data type)
12. Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword
13. Write a program to show the use of static functions and to pass variable length arguments in a function.
14. Write a program to demonstrate the concept of boxing and unboxing.
15. Create a multi-file program where in one file a string message is taken as input from the user and the function to display the message on the screen is given in another file (make use of Scanner package in this program).
16. Write a program to create a multilevel package and also creates a reusable class to generate Fibonacci series, where the function to generate fibonacii series is given in a different file belonging to the same package.
17. Write a program that creates illustrates different levels of protection in classes/subclasses belonging to same package or different packages
18. Write a program ―DivideByZero‖ that takes two numbers a and b as input, computes a/b, and invokes Arithmetic Exception to generate a message when the denominator is zero.
19. Write a program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.
16
20. Write a program to create your own exception types to handle situation specific to your application (Hint: Define a subclass of Exception which itself is a subclass of Throwable).

21. Write a program to demonstrate priorities among multiple threads.
22. Write a program to demonstrate multithread communication by implementing synchronization among threads (Hint: you can implement a simple producer and consumer problem).
23. Write a program that creates a Banner and then creates a thread to scrolls the message in the banner from left to right across the applet,,s window.
24. Write a program to get the URL/location of code (i.e. java code) and document(i.e. html file).
25. Write a program to demonstrate different mouse handling events like mouseClicked(), mouseEntered(), mouseExited(), mousePressed, mouseReleased() and mouseDragged().
26. Write a program to demonstrate different keyboard handling events.
27. Write a program to generate a window without an applet window using main() function.
28. Write a program to demonstrate the use of push buttons.

Reference Books

1. Ken Arnold, James Gosling, David Homes, "The Java Programming Language", 4th Edition, 2005.
2. James Gosling, Bill Joy, Guy L Steele Jr, GiladBracha, Alex Buckley"The Java Language Specification, Java SE 8 Edition (Java Series)", Published by Addison Wesley, 2014.
3. Joshua Bloch, "Effective Java" 2nd Edition,Publisher: Addison-Wesley, 2008.
4. Cay S. Horstmann, GaryCornell, "Core Java 2 Volume 1 ,9th Edition,Printice Hall.2012
5. Cay S. Horstmann, Gary Cornell, "Core Java 2 Volume 2 - Advanced Features)", 9th Edition, Printice Hall.2013
6. Bruce Eckel, "Thinking in Java", 3rd Edition, PHI, 2002.
7. E. Balaguruswamy, "Programming with Java", 4th Edition, McGraw Hill.2009.
8. Paul Deitel, Harvey Deitel, "Java: How to Program", 10th Edition, Prentice Hall, 2011.
9. "Head First Java", Orielly Media Inc. 2nd Edition, 2005.
10. David J. Eck, "Introduction to Programming Using Java", Published by CreateSpace Independent Publishing Platform, 2009.
11. John R. Hubbard, "Programming with JAVA", Schaum's Series, 2nd Edition

Major/DS Course (Core):

COMP 4013: Database Management Systems (Theory)

Credit: 4 **60 Hrs**

Course Objective

The course introduces the students to the fundamentals of database management system and its architecture. Emphasis is given on the popular relational database system including data models and data manipulation. Students will learn about the importance of database structure and its designing using conceptual approach using Entity Relationship Model and formal approach using Normalization. The importance of file indexing and controlled execution of transactions will be taught. The course would give students hands-on practice of structured query language in a relational database management system and glimpse of basic database administration commands.

Course Learning Outcomes

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

1. Use database management system software to create and manipulate the database.
2. Create conceptual data models using entity relationship diagrams for modeling real-life situations and designing the database schema.
3. Use the concept of functional dependencies to remove redundancy and update anomalies.
4. Apply normalization theory to get a normalized database scheme.
5. Write queries using relational algebra, a procedural language.
6. Implement relational databases and formulate queries to get solutions of a broad range of data retrieval and data update problems using SQL .
7. Write and execute SQL queries through a high-level language via ODBC connection
8. Database administration commands such as creating/removing users, granting/revoking different privileges to the database users; creating assertions, triggers, and indexes.
9. Learn the importance of index structures and concurrent execution of transactions in database systems.

Syllabus

1.Introduction (6 Hrs)

Characteristics of database approach, data models, database system architecture and data independence.

2.Entity Relationship (ER) Modeling (8 Hrs)

Entity types, relationships, constraints.

3.Relation data model (20 Hrs)

Relational model concepts, relational constraints, relational algebra, SQL queries

4. Database design (15 Hrs)

Mapping ER/EER model to relational database, functional dependencies, Lossless decomposition,

Normal forms (upto BCNF).

5. Transaction Processing (3 Hrs)

ACID properties, concurrency control & recovery

6. File Structure and Indexing (8 Hrs)

Operations on files, File of Unordered and ordered records, overview of File organizations, Indexing structures for files(Primary index, secondary index, clustering index), Multilevel indexing using B and B+ trees.

Books Recommended:

1. R. Elmasri, S.B. Navathe, Fundamentals of Database Systems 6th Edition, Pearson Education, 2010.
2. R. Ramakrishanan, J. Gehrke, Database Management Systems 3rd Edition, McGraw-Hill, 2002.
3. Silberschatz, H.F. Korth, S. Sudarshan, Database System Concepts 6th Edition, McGrawHill, 2010.
4. R. Elmasri, S.B. Navathe Database Systems Models, Languages, Design and application Programming, 6th Edition, Pearson Education, 2013.

Database Management Systems Lab
Credit: 1 **30Hrs**

Create and use the following database schema to answer the given queries.
EMPLOYEE Schema
Field Type NULL KEY DEFAULT
Eno Char(3) NO PRI NIL
Ename Varchar(50) NO NIL
Job_type Varchar(50) NO NIL
Manager Char(3) Yes FK NIL
Hire_date Date NO NIL
Dno Integer YES FK NIL
Commission Decimal(10,2) YES NIL
Salary Decimal(7,2) NO NIL
DEPARTMENT Schema
Field Type NULL KEY DEFAULT
Dno Integer No PRI NULL
Dname Varchar(50) Yes NULL
Location Varchar(50) Yes New Delhi
Query List
1. Query to display Employee Name, Job, Hire Date, Employee Number; for each employee with the Employee Number appearing first.
2. Query to display unique Jobs from the Employee Table.
3. Query to display the Employee Name concatenated by a Job separated by a comma.
4. Query to display all the data from the Employee Table. Separate each Column by a comma and name the said column as THE_OUTPUT.
5. Query to display the Employee Name and Salary of all the employees earning more than $2850.
6. Query to display Employee Name and Department Number for the Employee No= 7900.
7. Query to display Employee Name and Salary for all employees whose salary is not in the range of $1500 and $2850.
8. Query to display Employee Name and Department No. of all the employees in Dept 10 and Dept 30 in the alphabetical order by name.
9. Query to display Name and Hire Date of every Employee who was hired in 1981.
10. Query to display Name and Job of all employees who don,,t have a current Manager.

11. Query to display the Name, Salary and Commission for all the employees who earn commission.
12. Sort the data in descending order of Salary and Commission.
13. Query to display Name of all the employees where the third letter of their name is _A,,.

14. Query to display Name of all employees either have two _R,,s or have two _A,,s in their name and are either in Dept No = 30 or their Manger,,s Employee No = 7788.
15. Query to display Name, Salary and Commission for all employees whose Commission Amount is 14 greater than their Salary increased by 5%.
16. Query to display the Current Date.
17. Query to display Name, Hire Date and Salary Review Date which is the 1st Monday after six months of employment.
18. Query to display Name and calculate the number of months between today and the date each employee was hired.
19. Query to display the following for each employee <E-Name> earns < Salary> monthly but wants < 3 * Current Salary >. Label the Column as Dream Salary.
20. Query to display Name with the 1st letter capitalized and all other letter lower case and length of their name of all the employees whose name starts with _J,,, ,,A,, and _M,,.
21. Query to display Name, Hire Date and Day of the week on which the employee started.
22. Query to display Name, Department Name and Department No for all the employees.
23. Query to display Unique Listing of all Jobs that are in Department # 30.
24. Query to display Name, Dept Name of all employees who have an _A,, in their name.
25. Query to display Name, Job, Department No. And Department Name for all the employees working at the Dallas location.
26. Query to display Name and Employee no. Along with their Manger,,s Name and the Manager,,s employee no; along with the Employees,, Name who do not have a Manager.
27. Query to display Name, Dept No. And Salary of any employee whose department No. and salary matches both the department no. And the salary of any employee who earns a commission.
28. Query to display Name and Salaries represented by asterisks, where each asterisk (*) signifies $100.
29. Query to display the Highest, Lowest, Sum and Average Salaries of all the employees
30. Query to display the number of employees performing the same Job type functions.
31. Query to display the no. of managers without listing their names.
32. Query to display the Department Name, Location Name, No. of Employees and the average salary for all employees in that department.
33. Query to display Name and Hire Date for all employees in the same dept. as Blake.
34. Query to display the Employee No. And Name for all employees who earn more than the average salary.
35. Query to display Employee Number and Name for all employees who work in a department with any employee whose name contains a _T.
36. Query to display the names and salaries of all employees who report to King.
37. Query to display the department no, name and job for all employees in the Sales Department.

Minor Course:

COMP 4021 :Programming in C
(For other allied discipline)

Credit :3 **45 Hrs**

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. Also by learning the basic programming constructs they can easilyswitch 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:

Introduction: Basic Structure, Character sets, Keywords, Identifiers, Constants, Variables, Data Types, ProgramStructure. Operators: Arithmetic, Relational, Logical and Assignment.(15 Hrs) 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)

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+……
4. WAP to compute the sum of the first n terms of the following series S =1-2+3-4+5…….
5. Write a function that checks whether a given string is Palindrome or not. Use this function to findwhether 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 primenumbers 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 alsoinclude 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 ascommand 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 andcircumference from the main() function.

15. Write a menu driven program to perform following operations on strings:
    - Show address of each character in string
    - Concatenate two strings without using strcat function.
    - Concatenate two strings using strcat function.
    - Compare two strings
    - Calculate length of the string (use pointers)
    - Convert all lowercase characters to uppercase
    - Convert all uppercase characters to lowercase
    - Calculate number of vowels
    - Reverse the string

16. Given two ordered arrays of integers, write a program to merge the two-arrays to get an orderedarray.
17. WAP to display Fibonacci series (i)using recursion, (ii) using iteration
18. WAP to calculate Factorial of a number (i)using recursion, (ii) using iteration
19. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion.
20. Create Matrix class using templates. Write a menu-driven program to perform following Matrix

operations (2-D array implementation):

21. Sum b) Difference c) Product d) Transpose

22. Write a menu-driven program, using user-defined functions to find the area ofrectangle, square, circle and triangle by accepting suitable input parameters from user.

23. WAP to displaythe first n terms of Fibonacci series.

24. WAP to find factorial of the given number.

25. WAP to find sum of the following series for n terms: 1 – 2/2! + 3/3! --------- n/n!

26. WAP to calculate the sum and product of two compatible matrices.

Semester- V

Major/DS Course (Core): COMP 5011 (Theory)- Data Communications & Networking

Credit: 04 **60 Hours**

Course Objective

The course objective of this course is to understand the basic concepts behind computer networks and data communication. It helps students to learn the use of different layers in standard reference models used for communication. It also assists students to know main features of protocols used at various layers and understand the utility of different networking devices.

Course Learning Outcomes

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

i. Describe the hardware and software components used in a network.

ii. Compare OSI and TCP/IP reference models at various layers.

iii. Describe, analyse and compare different data link, network, transport and application layer protocols.

iv. Design/implement data link and network layer protocols in a simulated networking environment.

Syllabus:

1. Data Communication Fundamentals and Techniques (15 Hours)

Introduction: Communication systems, Analogue data, digital data, Communication channels, Synchronous data, Asynchronous data.

Analog and digital signal; data-rate limits; digital to digital line encoding schemes; pulse code

modulation; parallel and serial transmission, digital to analog modulation , ASK, FSK, PSK, Quadrature PSK, Differential PSK. multiplexing techniques- FDM, TDM;

Transmission media: Twin wire, Coaxial cable, Radio, VHF and microwaves, satellite communication, Optical fiber.

2. Introduction to Computer Networks (5 Hours)

Network definition; network topologies; network classifications; network protocol; layered

Network architecture; overview of OSI reference model; overview of TCP/IP protocol suite.

3. Network Switching Techniques and Access mechanisms (10 Hours)

Circuit switching; packet switching- connectionless datagram switching, connection-oriented

Virtual circuit switching; dial-up modems; digital subscriber line; cable TV for data transfer.

4. Data Link Layer Functions and Protocol (10 Hours)

Error detection and error correction techniques (CRC, Hamming-codes); data-link control-

Framing and flow control; error recovery protocols- stop and wait ARQ, go-back-n ARQ; Point

to Point Protocol on Internet.

5. Multiple Access Protocol and Networks (5 Hours)

ALOHA & CSMA protocols; CDMA; Ethernet LANs; connecting LAN and back-bone

networks- repeaters, hubs, switches, bridges, router and gateways;

6. Networks Layer Functions and Protocols (5 Hours)

Routing; static and dynamic routing algorithms; network layer protocol of Internet- IP protocol,

Internet control protocols.

7. Transport Layer Functions and Protocols (5 Hours)

Transport services- error and flow control, Connection establishment and release- three way

handshake, TCP and UDP.

8. Overview of Application layer protocol (5 Hours)

Overview of DNS protocol; overview of WWW & HTTP protocol.

Reference Books

1. B. A. Forouzan: Data Communications and Networking, Fourth edition, THM ,2007.

2. A. S. Tanenbaum: Computer Networks, Fourth edition, PHI , 2002

COMP 5011 (Practical): Computer Networks

Credit:1 **30 Hours**

All programs should be developed in C / Java / Python

1. Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.

2. Simulate Hamming-code based error detection & correction algorithm for noisy channel.

3. Simulate and implement stop and wait protocol for noisy channel.

4. Simulate and implement go back N sliding window protocol.

5. Simulate and implement selective repeat sliding window protocol.

6. Simulate and implement MST construction (Prim‟s, Kruskal‟s) for Ethernet

7. Simulate and implement the various routing algorithms (RIP, Distance-Vector routing, Dijkstra‟s, Bellman-Ford, Floyd-Warshall, Flooding).

Major/DS Course (Core): COMP 5012 (Theory)- Computer Organization

Credit: 04 **60 Hours**

Course Objective

This course introduces the students to the fundamental concepts of digital computer organization, design and architecture. It aims to develop a basic understanding of the building blocks of the computer system and highlights how these blocks are organized together to architect a digital computer system.

Course Learning Outcomes

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

i. Explain instruction cycle, pipelining and interrupts.

ii. Explain data communication between CPU, memory and I/O devices.

iii. Simulate the design of a basic computer using a software tool.

Syllabus:

Introduction: Evolution of Computers, Stored program concept and Von-Neumann architecture, Information representation and codes, Building blocks of Computers. [10 Hours]

Register Transfer and micro operations: Concepts of bus, Data movement among registers, A language to represent conditional data transfer, Data movement from/to memory, Arithmetic and logical operations with register transfer, Timing in register transfer. [15 Hours]

CPU Architecture: Instruction format, Instruction execution, Fetch and execution cycles, CPU organization with large registers, Stacks and handling of interrupts and subroutines, Instruction pipelining: stages, hazards and methods to remove hazards. [15 Hours]

Micro-programmed control unit: Basic organization of micro-programmed controller, Horizontal and vertical formats, Address sequencer. [5 Hours]

I/O Organization: Strobe based and handshake-based communication, Vector and Priority interrupt, DMA based transfer. [5 Hours]

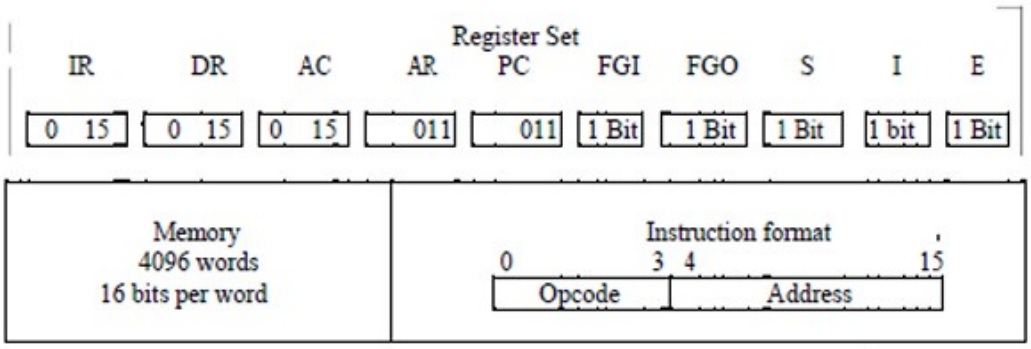
Programming the Basic Computer: Instruction formats, addressing modes, instruction

codes, machine language, assembly language, input output programming. [10 Hours].

COMP 5012 (Practical): Computer Organization

Credit:01 **30 Hours**

1. Create a machine based on the following architecture:



Basic Computer Instructions

Memory Reference			Register Reference		Input-Output		
Symbol	Hex		Symbol	Hex	Symbol	Hex	
AND	0xxx		CLA	E800	INP	F80 0	
ADD	2xxx		CLE	E400	OUT	F40 0	
LDA	4xxx	Direct	CMA	E200	SKI	F20 0	
STA	6xxx	Addressing	CME	E100	SKO	F10 0	
BUN	8xxx		CIR	E080	ION	F08 0	Optional
BSA	Axxx		CIL	E040	IOF	F04 0	
ISZ	Cxxx		INC	E020			
AND_I	1xxx		SPA	E010			
ADD_I	3xxx		SNA	E008			
LDA_I	5xxx	Indirect	SZA	E004			
STA_I	7xxx	Addressing	SZE	E002			
BUN_I	9xxx		HLT	E001			
BSA_I	Bxxx						
ISZ_I	Dxxx						

Refer to Chapter-5 of Morris Mano for description of instructions.

2) Create the micro operations and associate with instructions as given in the chapter (except interrupts). Design the register set, memory and the instruction set. Use this machine for the assignments of this section.

3) Create a Fetch routine of the instruction cycle.

4) Simulate the machine to determine the contents of AC, E, PC, AR and IR registers in hexadecimal after the execution of each of following register reference instructions:

a. CLA b. CLE c. CMA d. CME e. CIR f. CIL g. INC h. SPA i. SNA j. SZA k. SZE l. HLT

Initialize the contents of AC to (A937)16, that of PC to (022)16 and E to 1.

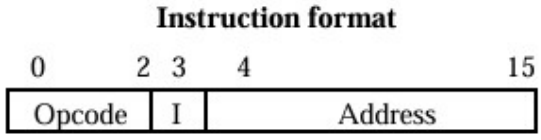
5. Simulate the machine for the following memory-reference instructions with I= 0 and address part = 082. The instruction to be stored at address 022 in RAM. Initialize the memory word at address 082 with the operand B8F2 and AC with A937. Determine the contents of AC, DR, PC, AR and IR in hexadecimal after the execution.

a. ADD b. AND c. LDA d. STA e. BUN f. BSA g. ISZ

6. Simulate the machine for the memory-reference instructions referred in above question with I= 1 and address part = 082. The instruction to be stored at address 026 in RAM. Initialize the

memory word at address 082 with the value 298. Initialize the memory word at address 298 with operand B8F2 and AC with A937. Determine the contents of AC, DR, PC, AR and IR in hexadecimal after the execution.

7. Modify the machine created in Practical 1 according to the following instruction format:



a. The instruction format contains a 3-bit opcode, a 1-bit addressing mode and a 12-bit address. There are only two addressing modes, I = 0 (direct addressing) and I = 1 (indirect addressing).

b. Create a new register I of 1 bit.

c. Create two new microinstructions as follows :

i. Check the opcode of instruction to determine type of instruction (Memory Reference/Register Reference/Input-Output) and then jump accordingly.

ii. Check the I bit to determine the addressing mode and then jump accordingly.

Major/DS Course (Core): COMP 5013 (Theory) - Software Engineering

Credit: 04 **60 Hours**

Course Objective

This course will provide fundamental approaches and techniques used to develop good quality software. This includes learning of various software development process frameworks, requirement analysis, design modelling, qualitative and quantitative software metrics, risk management, and testing techniques.

Course Learning Outcomes

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

i. Understand the software development models.

ii. Analyse and model customer requirements; build design models.

iii. Estimate and prepare schedule for software projects.

iv. Analyse the impact of risks involved in software development.

v. Design and build test cases, and to perform software testing.

Syllabus:

Software Engineering Fundamentals: Definition of software product, Software Engineering Paradigms; Software engineering, Knowledge engineering, and End user development approaches. [5 Hours]

System Analysis: An abstraction, Partitioning and projection, Systems specification, Software Requirements Specification (SRS) standards, Formal Specification methods, Specification tools, Flow based, Data based and Object –Oriented Analysis. [7 Hours]

System Documentation: Principles of system documentation, types of documentation and their importance [5 Hours]

System Planning: Data and fact gathering techniques-Interviewing, communications, presentations and site visit. Feasibility study, feasibility reports, prototyping, cost-benefit analysis-tools and techniques. [5 Hours]

Systems Design:

Idealized and constrained design, Process oriented design (Game and Sarson and Yourdon notations), Data oriented design (Warnier–Orr, E-R modelling), Object oriented design (Booch approach), Cohesion and Coupling, Design matrices, Design documentation standard. [5 Hours]

Role of CASE Tools: Relevance of CASE Tools, High-end and Low-end CASE Tools [5 Hours]

Coding and Programming : Choice of programming languages, Mixed language programming and cell semantics ,Re-engineering legacy systems, Coding standard. [5 Hours]

Software Quality and testing: Software quality assurance .Types of Software Testing (White Box and Black Box Testing, Unit Testing, Integration Testing, Verification and Validation of Software) , Debugging and Software Reliability analysis , Software quality and matrices, Software maturity model and extensions. [8 Hours]

Software Cost and Time estimation: Functions points , Issues in software cost estimation : Introduction to the Rayleigh curve, Algorithmic cost models (COCOMO, Putnam- Slim, Watson, and Felix), Other approaches to software cost and Size estimation (software complexity, Delphi , costing by analogy). [8 Hours]

Software Project Management : Planning software , projects, Work breakdown structures, Integrating software design and project planning ,Software project teams, Projecting monitoring and control. [7 Hours]

Books

1. Software Engineering, A practioner's Approach- R. S. Pressman (Mc-Graw Hill Inc)

2. An Integrated Approach to Software Engineering-P.Jalote (Narosa Publication House)

3. R. Mall, Fundamentals of Software Engineering (2nd Edition), Prentice-Hall of India,

2004.

COMP 5013 (Tutorial) : Software Engineering

Tutorial: 15 Lectures

Tutorial based on theory classes.

Credit: 1

Semester- VI

Major/DS Course (Core) COMP 6011 (Theory): Numerical Methods

Credit: 03 **45 Hours**

Course Objective

i. To understand the Importance of error analysis and their propagation.

ii. To introduce the basic concepts of solving algebraic, transcendental equations and system of linear and non-linear equations.

iii. To understand techniques of interpolation and polynomial fitting.

iv. To understand methods too numerical differentiation and integration.

v. To understand numerical solution of ordinary differential equations.

vi. To understand basic concepts of probability theory and distributions.

Course Outcomes:

After completion of the course students shall be able to:

i. Calculate errors induced in the values by truncation of a series expansion.

ii. Find roots of linear and non-linear system (algebraic and transcendental) equations.

iii. Fit polynomials to a given set of data points.

iv. Solve differential and integral equations numerically.

Syllabus:

Floating point representation and computer arithmetic, Significant digits, Errors: Round-off error, Local truncation error, Global truncation error, Order of a method, Convergence and terminal conditions, Efficient computations [10 Hours]

Bisection method, Secant method, Regula−Falsi method [3 Hours]

Newton−Raphson method, Newton„s method for solving nonlinear systems [3 Hours]

Gauss elimination method (with row pivoting) and Gauss−Jordan method, Gauss Thomas method for tridiagonal systems [3 Hours]

Iterative methods: Jacobi and Gauss-Seidel iterative methods [3 Hours]

Interpolation: Lagrange’s form and Newton’s form Finite difference operators, Gregory Newton

forward and backward differences Interpolation ,Piecewise polynomial interpolation: Linear interpolation, Cubic spline interpolation (only method), Numerical differentiation: First derivatives and second order derivatives, Richardson extrapolation . [5 Hours]

Numerical integration: Trapezoid rule, Simpson's rule (only method), Newton−Cotes open

formulas Extrapolation methods: Romberg integration, Gaussian quadrature, Ordinary differential equation: Euler's method [8 Hours]

Modified Euler's methods: Heun method and Mid-point method, Runge-Kutta second methods: Heun method without iteration, Mid-point method and Ralston's method Classical 4th order Runge-Kutta method, Finite difference method for linear ODE . [10 Hours]

Reference Books:

[1] Laurence V. Fausett, Applied Numerical Analysis, Using MATLAB, Pearson, 2/e (2012)

[2] M.K. Jain, S.R.K. Iyengar and R.K. Jain, Numerical Methods for Scientific and Engineering

Computation, New Age International Publisher, 6/e (2012)

[3] Steven C Chapra, Applied Numerical Methods with MATLAB for Engineers and Scientists, Tata McGraw Hill, 2/e (2010)

[4] William N. Venables and David M. Smith, An Introduction to R. 2nd Edition. Network Theory Limited.2009

[5] Norman Matloff, The Art of R Programming - A Tour of Statistical Software Design, No Starch Press.2011

COMP 6011 (Practical): Numerical Methods

Credit:1 **30 Hours**

All programs should be developed using C / Java / Python

1. Find the roots of the equation by bisection method.

2. Find the roots of the equation by secant/Regula−Falsi method.

3. Find the roots of the equation by Newton's method.

4. Find the solution of a system of nonlinear equation using Newto's method.

5. Find the solution of tridiagonal system using Gauss Thomas method.

6. Find the solution of system of equations using Jacobi/Gauss-Seidel method.

7. Find the cubic spline interpolating function.

8. Evaluate the approximate value of finite integrals using Gaussian/Romberg integration.

9. Solve the boundary value problem using finite difference method.

Major/DS Course (Core): COMP 6012 (Theory)- Microprocessors

Credit: 03 **45 Hours**

Course Objective

This course introduces internal architecture, programming models of Intel Microprocessors (8085 - Pentium) and assembly language programming. Students will also learn interfacing of memory and I/O devices with microprocessors.

Course Learning Outcomes

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

i. Describe the internal architecture of Intel microprocessors.

ii. Define and implement interfaces between the microprocessor and the devices.

iii. Write assembly language programs

Syllabus:

(All concepts should be studied in the context of the Intel 8085 Microprocessor.)

Microprocessor architecture: Internal architecture, system bus architecture, memory and I/O interfaces (15 Hour)

Microprocessor programming: Register Organization, instruction formats, assembly language Programming (15 Hour)

Interfacing: Memory address decoding, cache memory and cache controllers, I/O interface, keyboard, display, timer, interrupt controller, DMA controller, video controllers, communication interfaces (15 Hour)

Recommended Books:

1. Ramesh Gaonkar: Microprocessor Architecture, Programming and Application with the 8085

2. Barry B. Brey : The Intel Microprocessors : Architecture, Programming and Interfacing. Pearson Education, Sixth Edition,2009.

3. Walter A Triebel, Avtar Singh; The 8088 and 8086 Microprocessors Programming, Interfacing, Software, Hardware, and Applications. PHI, Fourth Edition 2005.

COMP 6012 (Practical): Microprocessor

Credit :1 **30 Hours**

All programs should be developed for the Intel 8085 Microprocessor.

ASSEMBLY LANGUAGE PROGRAMMING

1. Write a program for 32-bit binary division and multiplication
2. Write a program for 32-bit BCD addition and subtraction
3. Write a program for Linear search and binary search.
4. Write a program to add and subtract two arrays
5. Write a program for binary to BCD conversion
6. Write a program for BCD to binary conversion

Major/DS Course (Core) : COMP 6013 (Theory)- Theory of Computation

Credit: 03 **45 Hours**

Course Objective

This course introduces formal models of computation, namely, finite automaton, pushdown automaton, and Turing machine; and their relationships with formal languages. Students will learn about the limitations of computing machines as this course addresses the issue of which problems can be solved by computational means (decidability vs undecidability). The course is aimed to -

i. Formalizes the notion of problems via formal languages

ii. Make students aware of the notion of computation using "abstract computing devices" called automata

iii. familiarize students with the hierarchy of classes of problems or formal languages (regular, context-free, context-sensitive, decidable, and undecidable) and the hierarchy of classes of automata (finite automata, pushdown automata, and Turing machines)

Course Learning Outcomes

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

i. Design a finite automaton, pushdown automaton or a Turing machine for a problem at hand.

ii. Apply pumping lemma to prove that a language is non-regular/non-context-free.

iii. Describe limitations of a computing machines and recognize what can be solved and what cannot be solved using these machines.

Syllabus:

1. Languages (5Hour)

Alphabets, string, language, Basic Operations on language, Concatenation, KleeneStar

2. Finite Automata and Regular Languages (15 Hour)

Regular Expressions, Transition Graphs, Deterministics and non-deterministic finite automata,

NFA to DFA Conversion, Regular languages and their relationship with finite automata, Pumping lemma and closure properties of regular languages, Moore & Mealy Machines.

3. Context free languages (15 Hour)

Context free grammars, parse trees, ambiguities in grammars and languages, Pushdown automata (Deterministic and Non-deterministic), Pumping Lemma, Properties of context free languages, normal forms.

4. Turing Machines and Models of Computations (10 Hour)

RAM, Turing Machine as a model of computation, Universal Turing Machine, Language

acceptability, decidability, halting problem, Recursively enumerable and recursive languages,

unsolvability problems.

Recommended Books:

1. Hoperoft, Aho, Ullman, Introduction to Automata theory, Language & Computation –3rd

Edition, Pearson Education. 2006

2. P. Linz, An Introduction to Formal Language and Automata 4th edition Publication Jones

Bartlett, 2006

3. Lewis & Papadimitriou, Elements of the theory of computation, PHI 1997.

4. Daniel I.A.Cohen, Introduction to computer theory, John Wiley, 1996

COMP 6013 (Tutorial): Theory of Computation

Tutorial: 15 Lectures

Tutorial based on theory classes.

Credit: 1

Major/DS Course (Core): COMP 6014 (Theory)- Artificial Intelligence

Credit: 03 **45 Hours**

Course Objective

The course objectives of this course are to:

Understand the foundations, basic concepts and techniques of Artificial Intelligence (AI).

Apply informed search techniques for different applications.

Impart knowledge about the use of core AI techniques, having applicability to a wide range of real-world problems.

Learn about various knowledge representation techniques and writing Prolog/LISP programs.

Learn about the latest techniques for developing AI systems.

Course Learning Outcomes

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

i. Identify problems that are amenable to solutions by specific AI methods.

ii. Appreciate the utility of different types of AI agents.

iii. Apply different informed search techniques for solving real world problems.

iv. Use knowledge representation techniques for AI systems.

v. Understand human level, data driven and end to end approaches to AI

Syllabus:

1. Introduction (5 Hours)

Introduction to Artificial Intelligence, Background and Applications, Turing Test and Rational

Agent approaches to AI, Introduction to Intelligent Agents, their structure, behavior and environment.

2. Problem Solving and Searching Techniques (10 Hour)

Problem Characteristics, Production Systems, Control Strategies, Breadth First Search, Depth First Search, Hill climbing and its Variations, Heuristics Search Techniques: Best First Search, A* algorithm, Constraint Satisfaction Problem, Means-End Analysis, Introduction to Game Playing, Min-Max and Alpha-Beta pruning algorithms.

3. Knowledge Representation (10 Hour)

Introduction to First Order Predicate Logic, Resolution Principle, Unification, Semantic Nets,

Conceptual Dependencies, Frames, and Scripts, Production Rules, Conceptual Graphs.

Programming in Logic (PROLOG) / LISP

4. Dealing with Uncertainty and Inconsistencies (10 Hour)

Truth Maintenance System, Default Reasoning, Probabilistic Reasoning, Bayesian Probabilistic Inference, Possible World Representations.

5. Understanding Natural Languages (10 Hour)

Parsing Techniques, Context-Free and Transformational Grammars, Recursive and Augmented

Transition Nets.

Books Recommended:

1. DAN.W. Patterson, Introduction to A.I and Expert Systems – PHI, 2007.
2. Russell &Norvig, Artificial Intelligence-A Modern Approach, LPE, Pearson Prentice Hall, 2nd edition, 2005.

3.Rich & Knight, Artificial Intelligence – Tata McGraw Hill, 2nd edition, 1991.

W.F. Clocksin and Mellish, Programming in PROLOG, Narosa Publishing

House, 3rd edition, 2001.

4.Ivan Bratko, Prolog Programming for Artificial Intelligence, Addison-Wesley, Pearson

Education, 3rd edition, 2000.

COMP 6014 (Practical): Artificial Intelligence

Credit: 1 **30 Hours**

All programs should be developed using Prolog or LISP

1. Write a prolog/lisp program to calculate the sum of two numbers.

2. Write a prolog/lisp program to find the maximum of two numbers.

3. Write a prolog/lisp program to calculate the factorial of a given number.

4. Write a prolog/lisp program to calculate the nth Fibonacci number.

5. Write a prolog/lisp program, insert n^{th}(item, n, into_list, result) that asserts that result is the list into_list with item inserted as the n„th element into every list at all levels.

6. Write a Prolog/lisp program to remove the n^{th} item from a list.

7. Write a Prolog/lisp program, remove-n^{th}(Before, After) that asserts the After list is the Before list with the removal of every n^{th} item from every list at all levels.

8. Write a Prolog/lisp program to implement append for two lists.

9. Write a Prolog/lisp program to implement palindrome (List).

10. Write a Prolog/lisp program to implement max(X,Y,Max) so that Max is the greater of two numbers X and Y.

11. Write a Prolog/lisp program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List.

12. Write a Prolog/lisp program to implement sumlist(List,Sum) so that Sum is the sum of a given list of numbers List.

13. Write a Prolog/lisp program to implement two predicates even length(List) and Odd length(List) so that they are true if their argument is a list of even or odd length respectively.

14. Write a Prolog/lisp program to implement reverse(List,ReversedList) that reverses lists.

15. Write a Prolog/lisp program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List using cut predicate.

16. Write a Prolog/lisp program to implement GCD of two numbers.

17. Write a prolog/lisp program that implements Semantic Networks/Frame Structures.