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.

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 candevelop 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	Course Type	Name of the Course	Credit	Lect.	Tuto.	Pract. /Viva	Full Marks	Distribution of Marks		
								Theory	Pract. / Tuto./ Viva-voce	Internal Assess ment
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 **… 3021**	Intermediate Level Course (Voc. Edn. & Trng.)	4				75			15
	Multi/Interdisciplinary **COMP 3031**	Introduction to Cyber Security (For other discipline)	3	3	0	0	50	40	NIL	10
	Ability Enhancement Course (AEC) [L$_1$-2 MIL] **… 3041**	Arabic/ Bengali/ Hindi/ Sanskrit/ Santali/ Urdu] or Equvlnt. Course from SWAYAM or UGC recognized others	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] **ENG 4041**	English or Equvlnt. Course from SWAYAM or UGC recognized others	2	2	0	0	50	40	0	10
	Total		**25**				**425**			

Students exiting the programme after securing 87 credits will be awarded UG Diploma in the relevant Discipline /Subject provided they secure additional 4 credits in skill based vocational courses offered during summer term of fourth semester.

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 30Hrs

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:

□ 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.

Software Lab Based on PHP:

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 : That 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

Guidelines for secure password and wi-fi security

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

Smartphone security guidelines

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

Cyber Security Initiatives in India

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

Cyber Security Assurance

Online Banking, Credit Card and UPI Security

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

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

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

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.

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 byuser 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 display the 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.