West Bengal State Council of Technical & Vocational Education and Skill Development
(Technical Education Division)

Syllabus
of

Diploma in Cyber Forensics & Information Security [CFS] Part-II

(4th Semester)

Revised 2022

WEST BENGAL STATE COUNCIL OF TECHNICAL & VOCATIONAL EDUCATION AND SKILL DEVELOPMENT

(A Statutory Body under West Bengal Act XXVI of 2013)
(Technical Education Division)
Karigari Bhavan, 4th Floor, Plot No. B/7, Action Area-III, Newtown, Rajarhat, Kolkata–700 160

WBSCTVESD Curriculum for Diploma Courses in Engineering and Technology

Semester – IV
(Cyber Forensics and Information Security Engineering)

4th Semester Cyber Forensics and Information Security Engineering							
			Hours per week				
Sl. No	Code No.	Course Title	L	T	P	Contact Hours	Credits
1.	CFS202	Python Programming	4	0	0	4	3
2.	CFS204	Introduction to DBMS	3	0	0	3	2.5
3.	CFS206	Introduction to Cyber Security	3	0	0	3	2.5
4.	CFS208	Cyber Security Laws, Standards and IPR	3	0	0	3	2
5.		Open Elective-1	3	0	0	3	2.5
6.	CFS214	Mini Project	0	0	3	3	2
7.	CFS216	Python Programming Lab	0	0	4	4	1.5
8.	CFS218	Introduction to DBMS Lab	0	0	4	4	1.5
9.	CFS220	Introduction to Cyber Security Lab	0	0	4	4	1.5
		Total	**16**	**0**	**15**	**31**	**19**

List of Open Elective-1 Courses [OE-1]

OE1-CFS210	Web Designing and Multimedia Technology
OE1-CFS212	Algorithms

Mini Project (specific areas of interest include (but are not limited to)):

- Web Designing and Multimedia Technology
- Techno-Legal Study of Cyber Crime
- Use case related to Cyber Security
- Use case related to Ethical Hacking

Detailed Curriculum Content for Semester-IV

Syllabus for Programming in Python (Theory)

Course Title	**Programming in Python**
Course Code: CFS 202	Semester: Fourth
Duration: Sixth Months	Maximum Marks:100
Teaching Scheme	Examination Scheme
Theory: 4 hrs./week	Mid Semester Test: 20 Marks, Quizzes, Viva-voce, Assignment: 10 Marks
Total hours: 64	Class Attendance: 10 Marks
Credit: 3	End Semester Exam.: 60 Marks
Pass Criterion: Students have to obtain at least 40% marks (pass marks) in both internal assessment and end semester examination separately.	
Pre-Requisites: Concepts of Programming Logic, Data Structures. Basic concepts of DBMS	

Course Objectives:

- To understand why Python is a useful scripting language for developers.
- To learn how to design and program Python applications.
- To learn how to use lists, tuples, and dictionaries in Python programs.
- To learn how to identify Python object types.
- To learn how to use indexing and slicing to access data in Python programs.
- To define the structure and components of a Python program.
- To learn how to write loops and decision statements in Python.
- To learn how to write functions and pass arguments in Python.
- To learn how to build and package Python modules for reusability.
- To learn how to read and write files in Python.
- To learn how to design object-oriented programs with Python classes.
- To learn how to use class inheritance in Python for reusability.
- To learn how to use exception handling in Python applications for error handling.

Course Outcomes: After completion of the course students will be able to:

- Describe the concepts of constants, variables, data types and operators.
- Develop programs using input and output operations.
- Write programs using different looping and branching statements.
- Write programs based on list, tuple, set and strings handling functions.
- Write programs using user-defined functions, structures and union.
- Write programs using python dictionary.
- Use database to store and access data.
- Use various data structures effectively in application programs.
- Trace and code recursive functions.
- Demonstrate understanding of various sorting algorithms, including bubble sort, insertion sort, selection sort, heap sort and quick-sort.
- Understanding of multi-threading.
- RegEx learning and programming

Course Content

Unit-1 6 hours

1. **Introduction to Python**
    1.1. What is Python and history of Python?
    1.2. Unique features of Python
    1.3. First Python Program
    1.4. Python Variables, Identifiers, Keywords and Indentation
    1.5. Comments and document interlude in Python
    1.6. Command line arguments
    1.7. Getting User Input
    1.8. Python Data Types
    1.9. Python Core objects and Functions
    1.10. Number and Maths

Unit-2 7 hours

2. **Lists, Ranges and Tuples in Python**
    2.1 Introduction
    2.2 Lists in Python
    2.3 More About Lists
    2.4 Understanding Iterators
    2.5 Generators, Comprehensions and Lambda Expressions
    2.6 Introduction
    2.7 Generators and Yield
    2.8 Next and Ranges
    2.9 Understanding and using Ranges
    2.10 More About Ranges
    2.11 Ordered Sets with tuples

Unit-3 3 hours

3. **Python Dictionaries and Sets**
    3.1 Introduction to the section
    3.2 Python Dictionaries
    3.3 More on Dictionaries
    3.4 Sets
    3.5 Python Sets Examples

Unit-4 8 hours

4. Python Data Structures

4.1 Arrays

4.2 Linked lists

4.3. Stacks (LIFOs)

- list: Simple, Built-In Stacks
- collections.deque: Fast and Robust Stacks
- queue.LifoQueue: Locking Semantics for Parallel Computing
- Stack Implementations in Python: Summary

4.4 Queues (FIFOs)

- list: Terribly Slow Queues
- collections.deque: Fast and Robust Queues
- queue.Queue: Locking Semantics for Parallel Computing
- multiprocessing.Queue: Shared Job Queues
- Queues in Python: Summary

4.5 Priority Queues

- list: Manually Sorted Queues
- heapq: List-Based Binary Heaps
- queue.PriorityQueue: Beautiful Priority Queues
- Priority Queues in Python: Summary

Unit-5 4 hours

5. Input and Output in Python

5.1 Reading and writing text files

5.2 Writing Text Files

5.3 Appending to Files and Challenge

5.4 Writing Binary Files Manually

5.5 Using Pickle to Write Binary Files

Unit-6 4 hours

6. Python built in functions

6.1 Python user defined functions

6.2 Python packages functions

6.3 Defining and calling Function

6.4 The anonymous Functions

6.6 Loops and statement in Python

6.7 Python Modules & Packages

Unit-7 4 hours

7. Object Oriented Concepts in Python

7.1 Overview of OOP

7.2 Creating Classes and Objects

7.3 Accessing attributes

7.4 Built-In Class Attributes

7.5 Destroying Objects

Unit-8 **4 hours**

8. **Python Exceptions Handling**
   - 8.1 What is Exception?
   - 8.2 Handling an exception
   - 8.3 try….except…else
   - 8.4 try-finally clause
   - 8.5 Argument of an Exception
   - 8.6 Python Standard Exceptions
   - 8.7 Raising an exceptions
   - 8.8 User-Defined Exceptions

Unit-9 **5 hours**

9. **Python Regular Expressions**
   - 9.1 What are regular expressions?
   - 9.2 The match Function
   - 9.3 The search Function
   - 9.4 Matching vs searching
   - 9.5 Search and Replace
   - 9.6 Extended Regular Expressions
   - 9.7 Wildcard

Unit-10 **4 hours**

10. Python Multithreaded Programming
   - 10.1 What is multithreading?
   - 10.2 Starting a New Thread
   - 10.3 The Threading Module
   - 10.4 Synchronizing Threads
   - 10.5 Multithreaded Priority Queue
   - 10.6 Python Spreadsheet Interfaces

Unit-11 **6 hours**

11. Using Databases in Python
   - 11.1 Database Access using Python
   - 11.2 Create Database Connection
   - 11.3 CREATE, INSERT, READ, UPDATE and DELETE Operation
   - 11.4 DML and DDL Operation with Databases
   - 11.5 Performing Transactions
   - 11.6 Handling Database Errors
   - 11.7 Web Scraping in Python

Unit-12 9 hours

12. Python for Data Analysis

12.1 Numpy:
12.2 Introduction to numpy
12.3 Creating arrays
12.4 Using arrays and Scalars
12.5 Indexing Arrays
12.6 Array Transposition
12.7 Universal Array Function
12.8 Array Processing
12.9 Array Input and Output
12.10 Pandas:
12.11 What is pandas?
12.12 Where it is used?
12.13 Series in pandas
12.14 Index objects
12.15 Reindex
12.16 Drop Entry
12.17 Selecting Entries
12.18 Data Alignment
12.19 Rank and Sort
12.20 Summary Statics
12.21 Missing Data
12.22 Index Hierarchy
12.23 Matplotlib: Python for Data Visualization
12.24 Welcome to the Data Visualization Section
12.25 Introduction to Matplotlib

Text books:

1. Core Python Programming, R. Nageswara Rao, Wiley India
2. Python: The Complete Reference, Martin C. Brown, McGraw-Hill
3. Let Us Python, Yashavant Kanetkar, BPB Publication
4. Introduction to Programming using Python, Y D. Liang, Pearson
5. Python Programming A Modular Approach, S. Thaneja, N. Kumar, Pearson

Reference books:

1. Python Programming: Using Problem Solving Approach, Reema Thareja.
2. Fundamentals of Python: Data Structures, Kenneth Lambert.
3. Python Data Science Handbook: Essential Tools for Working with Data, Jake Vanderplas

List of open Source software/learning Websites:

- https://docs.python.org/3/tutorial/
- https://www.w3schools.com/python/
- https://www.jetbrains.com/pycharm/download/#section=windows

Syllabus for Programming in Python Lab

Course Title	**Programming in Python Lab**				
Course Code: CFS 216	Semester: Fourth				
Duration:	Maximum Marks:100				
Teaching Scheme	Continuous Assessment-60			End Semester Assessment-40	
Practical: 4 hrs./week Total hours: 64	Assignments (to be allotted)	Class Performance	Class Attendance	Assignment on the day of Viva-voce	Viva-voce (Before Board of Examiners)
Credit: 1.5	30	20	10	20	20
Pass Criterion: Students have to obtain at least 40% marks (pass marks) in both continuous assessment and end semester Assessment separately.					
Pre-Requisites: Concepts of Programming Logic, Data Structures. Basic concepts of DBMS					

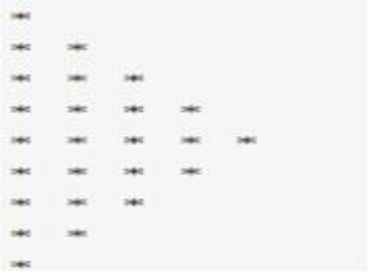
Skills to be developed:

- Use of python programming language constructs in program implementation.
- Apply different logics to solve given problems.
- Writing programs using different implementations for the same problem.
- Identify different types of errors as syntax, semantic, fatal, linker & logical.
- Debugging of programs.
- Understanding different steps to develop programs such as - Problem definition. Analysis, Design of logic, Coding, Testing.
- Maintenance (Modifications, error corrections, making changes etc.)

List of Practical:

First learn how to install Python package and Setup environment variables

1. Print the following pattern



```
*
*   *
*   *   *
*   *   *   *
*   *   *   *   *
*   *   *   *
*   *   *
*   *
*
```

2. Take, n float numbers as an input from the user, the find the average.
3. Count all lower case, upper case, digits, and special symbols from a given string.
4. Generate 100 random lottery tickets and pick two lucky tickets from it as a winner.

- The lottery number must be 10 digits long.
- All 100 ticket number must be unique.

5. Make a Search Engine Optimizer friendly URL from a topic name.

Input “A notes on Special theory of Relativity”
Output: https://example.com/a-notes-on-special-theory-of-relativity

6. Write a Python program to convert a given dictionary into a list of lists.

Original Dictionary:
{1: 'red', 2: 'green', 3: 'black', 4: 'white', 5: 'black'}
Convert the said dictionary into a list of lists:
[[1, 'red'], [2, 'green'], [3, 'black'], [4, 'white'], [5, 'black']]
Original Dictionary:
{'1': 'Austin Little', '2': 'Natasha Howard', '3': 'Alfred Mullins', '4': 'Jamie Rowe'}
Convert the said dictionary into a list of lists:
[['1', 'Austin Little'], ['2', 'Natasha Howard'], ['3', 'Alfred Mullins'], ['4', 'Jamie Rowe']]

7. Write a Python program to sort a list of elements using the merge sort algorithm. Note: According to Wikipedia "Merge sort (also commonly spelled merge sort) is an O(n log n) comparison-based sorting algorithm. Most implementations produce a stable sort, which means that the implementation preserves the input order of equal elements in the sorted output."

8. Write a Python program to sort a list of elements using the quick sort algorithm.
   Note: According to Wikipedia "Quicksort is a comparison sort, meaning that it can sort items of any type for which a "less-than" relation (formally, a total order) is defined. Inefficient implementations it is not a stable sort, meaning that the relative order of equal sort items is not preserved. Quicksort can operate in-place on an array, requiring small additional amounts of memory to perform the sorting."

9. Write a Python program for counting sort.
   According to Wikipedia "In computer science, counting sort is an algorithm for sorting a collection of objects according to keys that are small integers; that is, it is an integer sorting algorithm. It operates by counting the number of objects that have each distinct key value, and using arithmetic on those counts to determine the positions of each key value in the output sequence. Its running time is linear in the number of items and the difference between the maximum and minimum key values, so it is only suitable for direct use in situations where the variation in keys is not significantly greater than the number of items. However, it is often used as a subroutine in another sorting algorithm, radix sort that can handle larger keys more efficiently".

10. Write a Python program to print a given doubly linked list in reverse order.
11. Write a Python program to insert an item in front of a given doubly linked list.
12. Write a Python program to search a specific item in a given doubly linked list and return true if the item is found otherwise return false.
13. Write a Python program to delete a specific item from a given doubly linked list.
14. Write a Python function to check whether a number is perfect or not.
15. Write a Python function that prints out the first n rows of Pascal's triangle.
16. Write a Python program to execute a string containing Python code.
17. Write a Python program to calculate the harmonic sum of n-1.
    Note: The harmonic sum is the sum of reciprocals of the positive integers.
    Example :

$$1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\cdots$$

18. Write a Python program to find the greatest common divisor (GCD) of two integers.

19. Write a Python program that matches a word containing 'z', not at the start or end of the word.

20. Write a Python program to search the numbers (0-9) of length between 1 to 3 in a given string.

21. Write a Python program to read a random line from a file.
22. Write a Python program to assess if a file is closed or not.
23. Write a Python program to remove newline characters from a file.
24. Write a Python program that takes a text file as input and returns the number of words of a given text file. Note: Some words can be separated by a comma with no space.
25. Write a Python program to extract characters from various text files and puts them into a list.
26. Write a Python program to convert degree to radian. Note: The radian is the standard unit of angular measure, used in many areas of mathematics. An angle's measurement in radians is numerically equal to the length of a corresponding arc of a unit circle; one radian is just under 57.3 degrees (when the arc length is equal to the radius).
27. Write a Python program to convert radian to degree.
28. Write a Python program to calculate the area of a trapezoid.
    Note: A trapezoid is a quadrilateral with two sides parallel. The trapezoid is equivalent to the British definition of the trapezium. An isosceles trapezoid is a trapezoid in which the base angles are equal so.
29. Write a Python program to calculate the area of a parallelogram. Note: A parallelogram is a quadrilateral with opposite sides parallel (and therefore opposite angles equal). A quadrilateral with equal sides is called a rhombus, and a parallelogram whose angles are all right angles is called a rectangle.
30. Write a Python function student_data () which will print the id of a student (student_id). If the user passes an argument student_name or student_class the function will print the student name and class.
31. Write a Python program to create two empty classes, Student and Marks. Now create some instances and check whether they are instances of the said classes or not. Also, check whether the said classes are subclasses of the built-in object class or not.
32. Write a Python program to get the class name of an instance in Python.

Syllabus for Introduction to DBMS (Theory)

Course Title	**Introduction to DBMS (CFS)**
Course Code: CFS 204	Semester: Fourth
Duration: Six Months	Maximum Marks: 100
Teaching Scheme:	Examination Scheme:
Theory: 3 hrs./week	Mid Semester Test: 20 Marks Quizzes, Viva-voce, Assignment: 10 Marks
Credit : 2.5	Class Attendance: 10 Marks
Total hours: 48	End Semester Exam. : 60 Marks
Pass Criterion: Students have to obtain at least 40% marks (pass marks) in both internal assessment and end semester examination separately.	
Pre Requisites: Basic concepts of Programming Language. Understand the concepts of Computer System Organization and Operating Systems.	

Course Objectives:

- To study the development of database-driven applications using the capabilities provided by modern database management system software.
- To learn the concepts include conceptual modeling, relational database design and database query languages.
- To study and understand the basic concepts of RDBMS.
- To learn SQL and PL/SQL in detail.
- To learn how to work with any database.

Course Outcomes: After completion of the course students will be able to:

- Understand the concept of Database system and Client Server Architecture
- Understand and develop the concepts of Data Modeling, Security and Integrity.
- Understand and execute different SQL queries and PL/SQL programs.
- Normalize the database using normal forms.
- Understand the concept of query processing and Transaction processing.
- How to design a database, database-based applications

Course Content

Unit-1 10 hours

1. Database System Concepts and Data Modeling

1.1 Basic concepts, Advantages of a DBMS over file processing system, Data Abstraction, Database Languages, Data Independence.

1.2 Components of a DBMS and overall structure of a DBMS.

1.3 Data Models: Network Model, Hierarchical Model, E-R Model, Relational Model, Object Oriented Model

1.4 Client Server Architecture

1.5 E-R Model: Entities, Attribute, Relationship, Degree of Relationship, Mapping Cardinalities, Specialization, Aggregation

1.6 E-R / EER to Relational Model Mapping

Unit-2 6 hours

2. Relational Data Model and Security and Integrity Specification

2.1 Relational Model: Basic concepts, attributes and domains, Keys concept: Candidate key and Primary key, Integrity constraints: Domain, Entity Integrity constraints and On-delete cascade.

2.2 Security and Authorization.

2.3 Query Languages: Relational Algebra, Relational Calculus, Views.

Unit-3 11 hours

3. SQL and PL/SQL

3.1 Introduction to SQL queries, Creating, Inserting, Updating and deleting tables and using constraints, Set operations & operators, Aggregate functions, string functions and date, time functions, Null values, Nested sub-queries, Complex queries, Join concepts.

3.2 PL/SQL Introduction, PL/SQL block structure, variables, SQL statements in PL/SQL, PL/SQL control Structures, Cursors, Triggers, Functions, Packages, procedures.

3.3 Error handling in PL/ SQL

Unit-4 8 hours

4. Relational Database Design, Storage and File systems.

4.1 Purpose of Normalization, Data redundancy and updating anomalies, Functional Dependencies and Decomposition,

4.2 Process of Normalization using 1NF, 2NF, 3NF, BCNF, Multi-valued dependencies, Finding out candidate keys. Closure property and Armstrong’s axioms.

4.4 File Organization, Organization of records in files, Storage of Object Oriented databases, Basic concept of Indexing and Hashing.

4.5 Issues of efficiency of Queries and Optimization

Unit-5 9 hours

5. Query Processing and Transaction Processing

5.1 General strategies for query processing, Equivalence expressions, Selection & join operation.

5.2 Concept of transaction, ACID properties, States of transactions, Concurrent Executions, Serializability, Recoverability, Transaction definition in SQL.

Unit-6 4 hours

6. Case Study

6.1 Database Security and Forensics: Security Models for Database Applications, Database Auditing Models, Auditing Database Activities

Text and Reference Books:

1. Fundamentals of Database Systems, Elmasri & Navathe, Pearson Education
2. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata McGraw-Hill.
3. Database System Concepts, Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill, New Delhi, India.
4. Introduction to Database Systems, C .J. Date, Pearson Education
5. Introduction to SQL, Rick F. VanderLans, Pearson Education

Syllabus for Introduction to DBMS Lab

Course Title	**Introduction to DBMS Lab (CFS)**				
Course Code: CFS 218	Semester: Fourth				
Duration: Sixth Months	Maximum Marks:100				
Teaching Scheme	Continuous Assessment-60			End Semester Assessment-40	
Practical: 4 hrs./week Total hours: 64	Assignments (to be allotted)	Class Performance	Class Attendance	Assignment on the day of Viva-voce	Viva-voce (Before Board of Examiners)
Credit: 1.5	30	20	10	20	20
Pass Criterion: Students have to obtain at least 40% marks (pass marks) in both continuous assessment and end semester Assessment separately.					
Pre-Requisites: Basic concepts of Programming Language. Understand the concepts of Computer System Organization and Operating Systems.					

List of Practical:

1. Creating and Executing DDL in SQL.
2. Creating and Executing Integrity constraints in SQL.
3. Creating and Executing DML in SQL.
4. Executing relational, logical and mathematical set operators using SQL.
5. Executing group functions
6. Executing string operators & string functions.
7. Executing Date & Time functions.
8. Executing Data Conversion functions.
9. Executing DCL in SQL.
10. Executing Sequences and synonyms in SQL.
11. Execute thirty SQL queries (operators, functions, clauses, join concepts)
12. Program for declaring and using variables and constant using PL/SQL.
13. Program using if-then-else in PL/SQL
14. Program using for loop & while loop in PL/SQL.
15. Program using nested loop in Pl/SQL.
16. Write PL/SQL code for Error Handling using Built-in Exceptions
17. Cursor Manipulation: (a) Make use of Record Types (b) Processing Explicit Cursor (c) Use Cursor FOR Loops
18. Procedure: (a) Create Procedures (b) Passing Parameters IN, OUT and IN OUT with Procedures
19. Functions: (a) Create Stored Functions (b) Make use of Functions (c) Invoke Functions in SQL Statements
20. Triggers: (a) Use BEFORE and AFTER Triggers (b) Use different types of Triggers (ROW, STATEMENT, INSTEAD OF)

Text and Reference Books:

1. Introduction To SQL Mastering the RDB Language, Reck F. Van Der Lans, Pearson Publication
2. Oracle PL/SQL Programming, Steven Feuerstien, O’REILLY
3. Oracle PL?SQL by Example B. Rosenzweig & E. S. Rakhimov, Pearson Publication

Syllabus for Introduction to Cyber Security (Theory)

Course Title	**Introduction to Cyber Security (Theory)**
Course Code: CFS 206	Semester: Fourth
Duration: Sixth Months	Maximum Marks:100
Teaching Scheme	Examination Scheme
Theory: 3 hrs./week	Mid Semester Test: 20 Marks, Quizzes, Viva-voce, Assignment: 10 Marks
Total hours: 48	Class Attendance: 10 Marks
Credit: 2.5	End Semester Exam.: 60 Marks
Pass Criterion: Students have to obtain at least 40% marks (pass marks) in both internal assessment and end semester examination separately.	
Pre-Requisites: Basic Concepts of Computer System Organization. Basic knowledge of networking and Internet. Hands on Linux / Windows operating system.	

Course Objectives:

This course has been designed to provide the students an extensive overview of cyber security issues, tools and techniques that are critical in solving problems in cyber security domains. The course aims at providing students with concepts of computer security, cryptography, digital money, secure protocols, detection and other security techniques. The course will help students to gauge understanding in essential techniques in protecting Information Systems, IT infrastructure, analyzing and monitoring potential threats and attacks, devising security architecture and implementing security solutions. The students will also get a wider perspective to information security from national security perspective from both technology and legal perspective. So overall objective of this course focuses to cover all cyber security landscapes theoretically and practically.

Course outcomes: After completion of the course students will be able to:

- ❖ Understand, appreciate, employ, design and implement appropriate security technologies and policies to protect computers and digital information.
- ❖ Identify & Evaluate Information Security threats and vulnerabilities in Information Systems and apply security measures to real time scenarios
- ❖ Identify common trade-offs and compromises that are made in the design and development process of Information Systems
- ❖ Analyze and resolve security issues in networks and computer systems to secure an IT infrastructure.

Course Content

Unit-1 6 hours

1. Cyber Security Concepts

Essential Terminologies: CIA, Risks, Breaches, Threats, Vulnerability, Attacks, Exploits. Information Gathering (Social Engineering, Foot Printing and Scanning), Enumeration and System Hacking
Open Source Tools: Port Scanners, Network scanners like nmap, zenmap, Netcat

Unit-2 10 hours

2. Infrastructure and Network Security

Introduction to System Security, Server OS Security, Physical Security, Network packet Sniffing, Network Design Simulation. DOS/DDOS attacks. Man-in-the middle attack (Spoofing), Vulnerability Assessment, Penetration Testing, Asset Management. Intrusion detection and Prevention Techniques, Host based Intrusion prevention Systems, Security Information Management, Network Session Analysis.

Open Source Tools: DOS Attacks, DDOS attacks, Wireshark, Cain &Abel, Linux/Windows Firewall, Snort

Unit-3 14 hours

3. Cyber Security Vulnerabilities and Safe Guards

Internet Security, Concepts of Cloud Computing and Security, Social Network sites security, Cyber Security Vulnerabilities-Overview, vulnerabilities in software, System administration, Complex Network Architectures, Open Access to Organizational Data, Weak Authentication, Authorization, Unprotected Broadband Internet, Poor Cyber Security Awareness. Cyber Security Safeguards-Overview, Access control, Video Surveillance, Authentication. Open Web Application Security Project (OWASP), Concepts of IT Audit, Concepts of Web Site Audit and Vulnerabilities assessment, Popular Open Source Tools.

Unit-4 8 hours

4. Malware

Explanation of Malware, Types of Malware: Virus, Worms, Trojans, Rootkits, Robots, Adware, Spyware, Ransomware, Zombies etc.

OS Hardening (Process Management, Memory Management, Task Management, Windows Registry/ services configuration), Malware Analysis.

Open Source Tools: Antivirus Protection, Anti Spywares, System tuning tools, Anti-Phishing.

Unit-5 10 hours

5. Security in Evolving Technology

New Generation UTM with Sandboxing technique and AI analysis, Mobile Computing and Hardening on Android and iOS, Web server configuration and Security.

Deep Dive Security for HTTP Applications and Services. Basic Security for Web Services like SOAP, REST etc., Identity Management and Web Services, Identity Federation, Authorization Patterns, Security Considerations, Challenges, Popular Open Source Tools.

Text Books:

1. Nina Godbole, “Information System Security”, Wiley
2. Gupta Sarika, “Information and Cyber Security”, Khanna Publishing House, Delhi.
3. William Stallings, “Cryptography and Network Security”, Pearson Education/PHI
4. V.K. Jain, “Cryptography and Network Security”, Khanna Publishing House.
5. Atul Kahate, “Cryptography and Network Security”, McGraw Hill.
6. V.K. Pachghare, “Cryptography and Information Security”, PHI Learning
7. Bothra Harsh, “Hacking”, Khanna Publishing House, Delhi.

Reference Book:

1. Takedown by Tsutomu Shimomura, John Markoff
2. The Art of Deception: Controlling the Human Element of Security by Kevin D. Mitnick

Syllabus for Introduction to Cyber Security Lab

Course Title	**Introduction to Cyber Security Lab (Practical)**				
Course Code: CFS 220	Semester: Fourth				
Duration: Six Months	Maximum Marks:100				
Teaching Scheme	Continuous Assessment-60			End Semester Assessment-40	
Theory: 4 hrs./week Total hours: 64	Assignments (to be allotted)	Class Performance	Class Attendance	Assignment on the day of Viva-Voce	Viva-voce (Before Board of Examiners)
Credit: 1.5	30	20	10	20	20
Pass Criterion: Students have to obtain at least 40% marks (pass marks) in both continuous assessment and end semester Assessment separately.					
Pre Requisites: Basic knowledge of Computers, Basic knowledge of networking and Internet, Hands on Windows Operating System					

List of Practical:

1. Implementation to gather information from any PC's connected to the LAN using whois, port scanners, network scanning, Angry IP scanners etc.
2. Implementation of Symmetric and Asymmetric cryptography
3. Implementation of LSB Steganography
4. Implementation of MITM- attack using wireshark / network sniffers, Netcat, Nmap or Popular Open source tools
5. Implementation of Windows security using firewall and other tools
6. Implementation to identify web vulnerabilities, using OWASP project, Nikt0 or Popular Open source tools
7. Implementation of OS hardening and RAM dump analysis to collect the Artifacts and other information
8. Implementation of Cyber Forensics tools for Disk Imaging, Data acquisition, Data extraction and Data Analysis and recovery

Reference Books:

1. CompTIA CySA, Mike Chapple David Seidl, SYBEX
2. CYBER SECURITY ESSENTIALS, James Graham Richard Howard Ryan Olson, CRC Press

Syllabus for Cyber Security Laws, Standards and IPR (Theory)

Course Title	**Cyber Security Laws, Standards and IPR**
Course Code: CFS 208	Semester: Fourth
Duration: Sixth Months	Maximum Marks:100
Teaching Scheme	Examination Scheme
Theory: 3 hrs./week	Mid Semester Test: 20 Marks, Quizzes, Viva-voce, Assignment: 10 Marks
Total hours: 48	Class Attendance: 10 Marks
Credit: 2	End Semester Exam.: 60 Marks
Pass Criterion: Students have to obtain at least 40% marks (pass marks) in both internal assessment and end semester examination separately.	
Pre-Requisites: Concepts of Cyber Security and General Aptitude	

Course Objectives:

- Provide thorough understanding of Cyber Security Laws, Standards and IPR.

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

- Acquire good understanding of IT ACT and Programs.
- Acquire good understanding of Privacy acts and policy.
- Perform search and seizure of computers to obtain electronic evidence.
- Acquire good understanding of IPR.

Course Content

Unit-1 **5 hours**

1. **IT ACT and Programs**: Aims and Objectives; Overview of the Act; Jurisdiction; Electronic Governance; Electronic Evidence; Digital Signature Certificates; Digital signatures; Duties of Subscribers; Role of Certifying Authorities; The Cyber Regulations Appellate Tribunal; Internet Service Providers and their Liability; Powers of Police; Impact of the Act on other Laws; Social Networking Sites vis-à-vis Human Rights.

Unit-2 **5 hours**

2. **International Aspects of Cyber Crime:** International Perspectives; Budapest Convention on Cybercrime; ICANN’s core principles and the domain names disputes; Net neutrality; EU electronic communications regulatory framework; Web Content Accessibility Guidelines (WCAG).

Unit-3 5 hours

3. **Privacy acts and policy:** Introduction to Privacy; Personally Identifiable Information (PII);
4. Classification of privacy laws – Trespass, Negligence and Fiduciary; International legal standards on privacy - Asia-Pacific Economic Cooperation (APEC), Council of Europe, European Union (GDPR), OECD, United Nations; Privacy Laws of different countries; Privacy Policies and Management Standards, Data Protection Bill.

Unit-4 5 hours

1. **Searching and Seizing Computers and Obtaining Electronic Evidence:** Investigation of Cyber Crimes; Investigation of malicious applications; Agencies for investigation in India, their powers and their constitution as per Indian Laws; Procedures followed by First Responders; Search and Seizure Procedures of Digital Evidence; Securing the Scene, Documenting the Scene, Evidence Collection and Transportation - Data Acquisition, Data Analysis, Reporting; Digital Forensics - Computer Forensics, Mobile Forensics, Forensic Tools, Anti – Forensics.

Unit-5 4 hours

2. Critical Infrastructure Protection: Assets and Critical Infrastructure; Vulnerabilities; Threats; Attacks; Security Risk Assessment; Security Controls and Standards.

Unit-6 6 hours

3. **Electronic Commerce - Legal Issue:** E-Commerce; UNCITRAL Model; Legal aspects of E-Commerce; Digital Signatures; Technical and Legal issues; E-Commerce Trends and Prospects; E-taxation, E-banking, online publishing and online credit card payment; Employment Contracts; Non-Disclosure Agreements; Shrink Wrap Contract; Source Code; Escrow Agreements.

Unit-7 6 hours

4. **Legal Considerations in Designing and Implementing Electronic Processes:** Legal issues involved in Conversion to Electronic Processes; Availability of Information; Legal Sufficiency of Electronic Records; Reliability of Electronic Information; Legal Requirements Affecting Electronic Processes; Risk Assessment and Reduction.

Unit-8 8 hours

5. Intellectual Property: Cyber Law and IPRs; Understanding Copy Right in Information Technology; Software - Copyrights Vs Patents debate; Authorship and Assignment Issues; Copyright in Internet; Multimedia and Copyright issues; Software Piracy; Patents; European position on Computer related Patents; Legal position of U.S. on Computer related Patents; Indian position on Computer related Patents; Trademarks; Trademarks in Internet; Domain name registration; Domain Name Disputes & WIPO; Databases in Information Technology; Protection of databases; Position in USA, EU and India.

Unit-9 4 hours

6. **Case Studies:** Electronic / Digital Evidence laws & Case Laws - Indian & International Cases.

Text books:

1. Cyber Law & Cyber Crimes By Advocate Prashant Mali; Snow White publications, Mumbai
2. CYBER LAW & CYBER CRIMES SIMPLIFIED by Cyber Infomedia by Adv. Prashant Mali (Author)
3. Cyber Law in India by Farooq Ahmad; Pioneer Books
4. Information Technology Law and Practice by Vakul Sharma; Universal Law Publishing Co. Pvt. Ltd.
5. The Indian Cyber Law by Suresh T. Vishwanathan; Bharat Law House New Delhi
6. Guide to Cyber and E – Commerce Laws by P.M. Bukshi and R.K. Suri; Bharat Law House, New Delhi
7. Guide to Cyber Laws by Rodney D. Ryder; Wadhwa and Company, Nagpur
8. Scene of the Cybercrime: Computer Forensics Handbook by Syngress.
9. Security and Incident Response by Keith J. Jones, Richard Bejtloich and Curtis W. Rose
10. Introduction to Forensic Science in Crime Investigation by Dr. (Smt) Rukmani Krishnamurthy.

Reference books:

Cyber Forensics: A Field Manual for Collecting, Examining, and Preserving Evidence of Computer Crimes

Syllabus for Web Designing and Multimedia Technology

Course Title	**Web Designing and Multimedia Technology (Open Elective-1)**
Course Code: OE1-CFS 210	Semester: Fourth
Duration: Sixth Months	Maximum Marks:100
Teaching Scheme	Examination Scheme
Theory: 3 hrs./week	Mid Semester Test: 20 Marks, Quizzes, Viva-voce, Assignment: 10 Marks
Total hours: 48	Class Attendance: 10 Marks
Credit: 2.5	End Semester Exam.: 60 Marks
Pass Criterion: Students have to obtain at least 40% marks (pass marks) in both internal assessment and end semester examination separately.	
Pre-Requisites: Fundamental knowledge of Computer Science and strong foundation on Computer hardware and accessories	

Course Objectives:

- To develop the skill & knowledge of Web page design.
- To combine moving images, graphics, text, and sound in meaningful ways is one of most powerful aspects of computer technology and which is multimedia and animation.

Course Outcomes: The students will be able to:

- Design and develop powerful GUI based components.
- Visualize the basic concept of HTML & CSS.
- Integrate multimedia content in web page.
- Understand the concepts of text, image, audio and video media.

Course Content

Unit-1 **5 hours**

1. Web Design Principles

1.1. Basic principles involved in developing a web site
1.2. Planning process, Five Golden rules of web designing
1.3. Designing navigation bar
1.4. Page design
1.5 Home Page Layout
1.6. Design Concept
1.7. What is World Wide Web ?
1.8. Why create a web site and Web Standards.

Unit-2 **9 hours**

2. Working with HTML

2.1. What is HTML ?
2.2. HTML Documents
2.3. Basic structure of an HTML document
2.4. Creating an HTML document
2.5. Mark up Tags
2.6. Heading-Paragraphs
2.7. Line Breaks
2.8. HTML Tags
2.9. Introduction to elements of HTML
2.10. Working with Text
2.11. Working with Lists
2.12. Tables and Frames
2.13. Working with Hyperlinks and Images
2.14. Working with Forms and controls

Unit-3 **9 hours**

3. Concept of CSS

3.1. Creating Style Sheet
3.2. CSS Properties
3.3. CSS Styling (Background, Text Format, Controlling Fonts)
3.4. Working with block elements and objects
3.5. Working with Lists and Tables
3.6. CSS Id and Class
3.7. Box Model (Introduction, Border properties, Padding Properties, Margin properties)
3.8. CSS Advanced (Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class, Navigation Bar, Image Sprites, Attribute sector)
3.9. CSS Color
3.10. Creating page Layout and Site Designs

Unit-4 **4 hours**

4. Basics of Multimedia

4.1. Concept of Multimedia
4.2. Multimedia data stream
4.3. Hardware & Software requirement
4.4. Application of Multimedia
4.5. Steps of creating Multimedia presentation
4.6. Concept of Hypermedia and Hypertext.

Unit-5 **5 hours**

5. Text and Image

5.1. Types of text
5.2. ASCII codes
5.3. Unicode standards
5.4. OCR
5.5. Text file Formats
5.6. Types of images
5.7. Color and color models
5.8. Image file formats

Unit-6 8 hours

6. Audio and Video

6.1. Nature of sound wave
6.2. Musical sound and noise
6.3. Tone and note
6.4. Microphone, amplifier and speaker
6.5. Synthesizer
6.6. Musical Instrument Digital Interface (MIDI)
6.7. Audio file formats
6.8. Video frames and frame rates
6.9. Analog video
6.10. Digital video
6.11. CC Camera feeds
6.12. Video file formats

Unit-7 8 hours

7. **Compression**

7.1. CODEC

7.2. Types of Compression
7.3. Lossless/Statistical Compression techniques
7.4. GIF image coding standard
7.5. Lossy / Perceptual Compression techniques
7.6. JPEG image coding steps
7.7. MPEG Compression basics
7.8. MPEG-1 Audio & Video
7.9. MPEG-2 Audio & Video
7.10. Concept of MPEG-4

Text books:

1. Web Technology: A Developer's Perspective, N.P. Gopalan and J. Akilandeswari, PHI Learning, Delhi
2. Principles of Multimedia, Ranjan Parekh, Tata McGraw Hill

Reference books:

7. Web Technology: Theory and Practice, M. Srinivasan, Pearson
8. Computer Graphics Multimedia and Animation, Malay Kr Pakhira, PHI
9. Multimedia Systems, Bufford, Pearson

Syllabus for Algorithms

Name of the Course: **Algorithms**	
Course Code: OE1-CFS 212	Semester: Fourth
Duration: Six Months	Maximum Marks:100
Teaching Scheme	Examination Scheme
Theory: 3 hrs./week	Class Test: 20 Marks
Total hours: 48	Teachers Assessment: 10 Marks
Credit: 2.5	End Semester Exam.: 70 Marks
Pre Requisites: Basic concepts of Computer System Organization, Data Structures and Programming Logic	

Course Learning Objectives:

The objective of this course is to prepare the student with the algorithmic foundations of computing. A sound grasp of algorithms is essential for any computer science engineer. Almost all programming involves algorithms at some level.

Course outcomes:

The student should be able to design basic algorithms for sorting and searching. The student should be able to understand the basic notions of time and space complexity of algorithms. The student should be able to implement sorting, searching, tree and graph algorithms in a modern computer programming language.

Course Content

Unit-1 7 hours

1. **Introduction**
   - 1.1. What is an algorithm?
   - 1.2. Characteristics of algorithm
   - 1.3. Analysis of algorithm: Asymptotic analysis of complexity bounds – best, average and worst-case behavior
   - 1.4. Performance measurement
   - 1.5. Time and space trade-offs
   - 1.6. Analysis of recursive algorithms through recurrence relations: Substitution method, Recursion tree method and Masters' theorem
   - 1.7. Concept of Randomized Algorithms

Unit-2 8 hours

2. Elementary data structures

2.1. Linear data structures: stack, queue, lists
2.2. Non-linear data structures: graphs, trees (basic terminologies, binary trees)
2.3. Dictionaries: binary search trees
2.4. Hashing: open hashing (separate chaining), closed hashing (open addressing), Priority queues
2.5. Heaps, Heap Sort
2.6. Sets and disjoint set union: introduction, union and find operations

Unit-3 7 hours

3. Divide-and-Conquer

3.1. General method
3.2. Binary search
3.3. Finding the maximum and minimum
3.4. Merge sort
3.5. Quicksort
3.6. Performance analysis

Unit-4 9 hours

4. Brute Force and Exhaustive Search

4.1. Brute Force Approach: Selection Sort and Bubble, Sequential Search and Brute-Force String Matching
4.2. Exhaustive Search: Traveling Salesman Problem, Assignment Problem, Depth-First Search and Breadth-First Search

Unit-5 8 hours

5. Greedy method

5.1. Basic method, use, Examples
- 5.1.1. Prim’s Algorithm
- 5.1.2. Kruskal’s Algorithm
- 5.1.3. Dijkstra’s Algorithm
- 5.1.4. Fractional Knapsack Problem
- 5.1.5. Huffman Trees and Codes

Unit-6 3 hours

6. Dynamic Programming

6.1. Basic method, use, Examples

6.1.1. Matrix-chain multiplication

6.1.2. Floyd Warshall’s all pair shortest path Algorithm

6.1.3. 0/1 Knapsack problem

Unit-7 3 hours

7. Backtracking

7.1. Basic method, use, Examples:

7.1.1. 8-queen problem

7.1.2. Graph coloring problem

Unit-8 3 hours

8. NP-hard and NP-complete problems

8.1. P class

8.2. NP-hard class

8.3. NP-complete class

8.4. Circuit Satisfiability problem

Text and Reference Books:

1. Introduction to Algorithms, Cormen, Leiserson, Rivest and Stein. MIT Press
2. Fundamentals of Computer Algorithms, Ellis Horowitz, S Sahni, University Press
3. Algorithms, Sedgewick and Wayne, Pearson
4. Introduction to The Design and Analysis of Algorithms, Anany Levitin, Pearson
5. Introduction to Theory of Computation, Sipser Michael, Cengage Learnng.
6. Design & Analysis of Algorithms, Gajendra Sharma, Khanna Publishing House