**Debre Markos University**

**Institute of Technology**

**Department of Software Engineering**

**2nd year students 2nd semester course list**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| * **Year II Semester II** | | | | | | | |
| **Course Title** | **Course Code** | **CP** | **CrHr** | **Lec** | **Lab** | **Tut** | **H/S** |
| Data Structure and Algorithms | SEng2022 | 5 | 3 | 2 | 3 | 0 | 5 |
| Computer Organization and Architecture | SEng2032 | 5 | 3 | 2 | 0 | 3 | 5 |
| Operating Systems and Systems Programming | SEng2034 | 7 | 4 | 3 | 3 | 0 | 7 |
| Web Design and Programming I | SEng2074 | 5 | 3 | 2 | 3 | 0 | 5 |
| Object Oriented Programming | SEng2062 | 5 | 3 | 2 | 3 | 0 | 5 |
| Fundamentals of Networking | SEng2072 | 7 | 4 | 3 | 3 | 0 | 7 |
| **Total** | | **34** | **20** | **14** | **15** | **3** | **34** |

**Course Outline #1**

|  |  |
| --- | --- |
| **Course Title** | Data Structure and Algorithms |
| **Course Code** | SEng2022 |
| **CP** | 5 (2hr Lecture, 3hr Lab) |
| **Module Title** | Basics of Programming |
| **Module Number** | 02 |
| **Pre-requisites** | SEng2021 |
| **Year** | II |
| **Semester** | II |
| **Status of Course** | Compulsory |

**Course Description**

This course aims to introduce you some basic data structures and algorithms which are to be used as tools in designing solutions to problems. It will make students familiar with the specification, usage, implementation and analysis of these data structures and algorithms. Introduction of Data Structures (Linear, non-Linear Data Structures) mainly focused and Algorithm Analysis Concept, Measuring Complexity, Complexity of Algorithm Big-O Notation. Simple Sorting and Searching Algorithms (Bubble Sort, Insertion Sort, Selection Sort, Sequential Searching, Binary Searching.). Abstract Data Types, Structures, Pointers, Arrays, Linked Lists, Stacks, Queues, Trees, Graphs. Advanced Sorting and Searching Algorithms (Shell Sort, Quick Sort, heap Sort, Merge Sort, and Hashing). Laboratory exercises are dedicated to practice the basics on concepts on data structures like Abstract Data Types, Structures, Pointers, Arrays, Linked Lists, Stacks, Queues, Trees, and Graphs. Advanced Sorting and Searching Algorithms using C++ programming language.

**Learning Outcomes**

At the end of the course students will be able to

* Explain the basic techniques for the design and analysis of efficient Algorithm;
* Determine when and how to use the various data structures including Linked lists, Queues, Stacks, Binary trees, Search trees and Graphs
* Design algorithms to solve real-life problems using the tools introduced Analyze your solution, and efficiently implement your solution.
* Apply data structures and algorithms that are frequently used in information processing

**Course Contents**

|  |  |
| --- | --- |
| ***Date*** | **Topics to be Discussed** |
| **Week 1-2** | **Chapter 1:Review of C++ concepts**  1.1. Arrays  1.2. Structures  1.3. Functions  1.4. Pointers |
| **Week 3-4** | **Chapter 2: Complexity analysis**  2.1. Computational and asymptotic complexity  2.2. Big-O, Ω, Θ, little-o and OO notations  2.3. Common complexity classes  2.4. Best, average and worst case complexity  2.5. Amortized complexity |
| **Week 4-5** | **Chapter 3: Linked lists**  3.1. Singly linked lists  3.2. Doubly linked lists  3.3. Circular lists  3.4. Skip lists  3.5. Self-organizing lists  3.6. Sparse tables |
| **Week 6-7** | **Chapter 4: Stacks and queues**  4.1. Stacks  4.2. Queues  4.3. Deques  4.4. Priority queues |
| **Week 8-9** | **Chapter 5: Recursion**  5.1. Recursive definitions  5.2. Function calls and recursive implementation  5.3. Tail recursion  5.4. Nontail recursion  5.5. Indirect recursion  5.6. Nested recursion  5.7. Excessive recursion  5.8. Backtracking |
| **Week 10-11** | **Chapter 6: Simple Sorting and Searching Algorithms**  6.1. Searching Algorithm  6.1.1. Linear Search (Sequential Search)  6.1.2. Binary Search  6.2. Sorting Algorithms  6.2.1. Insertion Sort  6.2.2. Selection Sort  6.2.3. Bubble Sort  6.3. Efficient sorting algorithms |
| **Week 12-13** | **Chapter 7: Binary trees**  7.1. Trees, binary trees and binary search trees  7.2. Implementing binary trees  7.3. Searching a binary tree  7.4. Tree traversal  7.4.1. Breadth-first  7.4.2. Depth-first  7.4.3. Stack less depth-first  7.5. Insertion  7.6. Deletion  7.7. Balancing a tree  7.8. Self-adjusting trees  7.9. Heaps  7.10. Polish notation and expression trees |
| **Week 14-15** | **Chapter 8: Hashing**  8.1. Hash functions  8.2. Collision resolution |

**Summary of Teaching Learning Methods**

The teaching-learning methodology will be student-centered with appropriate guidance of instructor/s during the student’s activities .There will be Lecture, Demonstrations, Tutorials, Reading assignments and Group Discussions

**Assessment Methods:** - As per DMiT academic regulation

**References**

1. The main course text is ―Data Structures and Algorithms in C++‖ by A. Drozdek (Brooks/Cole, 2001). However, there are no copies of this book in the FBE library. If you can find a copy somewhere it is the best choice, as it covers everything that we will cover in this course.
2. There is one eBook available on the department intranet: “Data Structures and Algorithms with Object-Oriented Design Patterns in C++” by B. R. Preiss. this covers most topics in this course.
3. Unfortunately, there are no books in the FBE library that specifically deal with data structures and algorithms, but there are a number of books on C++ that will cover some parts of the course.

**Course Outline #2**

|  |  |
| --- | --- |
| **Course Title** | Computer Organization and Architecture |
| **Course Code** | SEng2032 |
| **CP** | 5(2hr Lecture, 3hr Tutorial) |
| **Module Title** | Computer Systems |
| **Module Number** | 03 |
| **Pre-requisites** | None |
| **Year** | III |
| **Semester** | I |
| **Status of Course** | Compulsory |

**Course Description**

This course introduces students to the fundamental computer organization and architecture concepts. It covers digital systems, data representation, common digital components, register transfer language and micro operations, basic computer organization and design, central processing unit, memory organization, input-output organization, and pipeline and vector processing; case study of at least two microprocessor families and other components of computing system.

**Course Goals or Learning Outcomes:**

By the end of this course, students will be able to:

* Describe the basic structure and operation of a digital computer
* Explain in detail the operation of the arithmetic unit including the algorithms & implementation of fixed-point multiplication & division. and floating-point addition, subtraction,
* Identify different ways of communicating with I/O devices and standard I/O interfaces.
* Describe different performance enhancement of computer architecture
* Identify the processes involved in the basic operations of CPU
* Understand basic concepts of circuits and their design

**Course contents**

|  |  |
| --- | --- |
| **Week/Date** | **TOPIC TO BE DISCUSSED** |
| **Week 1** | **Introduction to Digital Systems**  1.1 Basic Concepts of digital systems  1.2 Logic gates  1.3 Boolean Algebra  1.4 Combinational circuit  1.5 Flip Flops  1.6 Sequential circuits |
| Week 2 | **Data Representation**  2.1 DMiTs, bytes, and words  2.2 Numeric data representation and number bases  2.3 Fixed- and floating-point systems  2.4 Signed and twos-complement representations  2.5 Data types  2.6 Representation of nonnumeric data (character codes, graphical data)  2.7 Representation of records and arrays |
| **Week 3** | **Common Digital Components**  3.1 Integrated Circuits  3.2 Decoders, multiplexers and registers  3.3 Binary counters  3.4 Memory units |
| **Week 4** | **Register Transfer Language and Micro Operations**  4.1 Register Transfer Language  4.2 Bus and Memory Transfer |
| **Week 5** | **Register Transfer Language and Micro Operations**  5.1 Arithmetic and Logic Operations  5.2 Shift Micro operations |
| **Week 6** | **Basic Computer Organization and Design**  6.1 Instructional Code  6.2 Computer Registers  6.3 Computer Instructions  6.4 Timing and Control |
| **Week 7** | **Basic Computer Organization and Design**  7.1 Memory Reference Instructions  7.2 Design of Basic Computer  7.3 Design of accumulator Logic |
| **Week 8** | **Central Processing Unit**  8.1 General Register Organization  8.2 Stack Organization  8.3 Instruction Formats |
| **Week 9** | **Central Processing Unit**  9.1 Addressing modes  9.2 Data Transfer and Manipulation  9.3 Program Control  9.4 Characteristics of RISC and CISC |
| **Week 10** | **Memory Organization**  10.1 Memory Hierarchy  10.2 Main Memory  10.3 Cache memory  10.4 Mapping Functions |
| **Week 11** | **Memory Organization :** External Memory  11.1 Magnetic Disks  11.2 RAID Technology  11.3 Optical disks  11.4 Magnetic Tape, |
| **Week 12** | **Input-Output Organization**  12.1 Peripheral Devices  12.2 Input-Output Interface  12.3 Asynchronous Data Transfer  12.4 Mode of Transfers |
| **Week 13** | **Input-Output Organization**  13.1 Priority Interrupts  13.2 Direct Memory Access(DMA)  13.3 Input-Output Processor(IOC)  13.4 Serial Communication |
| **Week 14** | **Pipeline and Vector Processing**  14.1 Parallel Processing  14.2 Pipelining |
| **Week 15** | **Pipeline and Vector Processing**  15.1 Vector Processing  15.2 Multiprocessors |
| **Week 16** | FINAL EXAM |

**Summary of Teaching Learning Methods**

The learning–teaching methodology will be student-centered with appropriate guidance of instructor/s during the student’s activities .There will be Lecture, Demonstrations, Tutorials, Reading assignments and Group Discussions

**Summary of Assessment Methods:**

The course will be assessed using the different assessment methods like: Quizzes, Reading assessments, Assignments, and Final exam

**References**

Text book: William Stalling, Computer Organization and Architecture: Designing for Performance, 7th Edition, Prentice Hall, 2006

**Course Outline #3**

|  |  |
| --- | --- |
| **Course Title** | Operating Systems and System Programming |
| **Course Code** | SEng2034 |
| **CP** | 7(3hr Lecture, 3hr Lab ) |
| **Module Title** | Computer Systems |
| **Module Number** | 03 |
| **Pre-requisites** | None |
| **Year** | II |
| **Semester** | II |
| **Status of Course** | Compulsory |

**Course Description**

What is an Operating System, history of operating system, operating system zoo; process concept, process scheduling, inter-process communication; threads; CPU scheduling, basic concepts, scheduling criteria, scheduling algorithms; process synchronization, the critical section problem, semaphores, monitors, classical synchronization problems; deadlocks, avoidance, prevention, detection; memory management, physical and virtual memory, swapping, allocation, paging, segmentation; file systems, access methods, directory structure, file sharing and protection; security, authentication, intrusion detection, encryption.

**Course Goals or Learning Outcomes**

By the end of this course, students will be able to:

* Explain the objectives and functions of modern operating systems
* Describe the functions of a contemporary operating system with respect to convenience, efficiency, and the ability to evolve.
* Explain the different states that a task may pass through and the data structures needed to support the management of many tasks.
* Explain conditions that lead to deadlock.
* Compare and contrast the common algorithms used for both preemptive and non- preemptive scheduling of tasks in operating systems, such as priority, performance comparison, and fair-share schemes.
* Explain the concept of virtual memory and how it is realized in hardware and software

**Course contents**

|  |  |
| --- | --- |
| **Week/Date** | **TOPIC TO BE DISCUSSED** |
| **Chapter 1** | **Overview**  5.1 Role and purpose of operating systems  5.2 History of operating system development  5.3 Types of operating systems |
| **Chapter 2** | **Processes and process management**  5.4 The process concept  5.5 The threads concept  **Processes and process management (Cont)**  5.6 Inter-process communication  **Processes and process management**  5.7 Process scheduling  **Processes and process management**  5.8 Deadlock |
| **Chapter 3** | **Memory management**  5.9 Review of physical memory and memory management hardware  5.10Overlays, swapping, and partitions  **Memory management**  5.11 Paging and segmentation  5.12 Page placement and replacement policies  **Memory management**  5.13Working sets and thrashing  ***5.14***Caching |
| **Chapter 4** | **Device management**  5.15. Characteristics of serial and parallel devices  5.16. Abstracting device differences  **Device management**  5.17. Buffering strategies  5.18. Direct memory access  5.19. Recovery from failures |
| **Chapter 5** | **File systems**  5.20Fundamental concepts (data, metadata, operations, organization, buffering,  sequential vs. non-sequential files)  5.21Content and structure of directories  **File systems**  5.22File system techniques (partitioning, mounting and un mounting, virtual file systems)  5.23Memory-mapped files  **File systems**  5.24Special-purpose file systems  5.25Naming, searching, and access |
| **Chapter 6** | **Security and protection**  5.26. Overview of system security  5.27. Policy/mechanism separation Security methods and devices |
| **Security and protection**  5.28. Protection, access, and authentication  5.29. Models of protection  5.30. Memory protection  5.31. Encryption  5.32 . Recovery management |
| **System Programming part** | **UNIT I. INTRODUCTION**  System software and machine architecture - The Simplified Instructional Computer (SIC) - Machine architecture - Data and instruction formats - addressing modes - instruction sets - I/O and programming.  **UNIT II. ASSEMBLERS**  Basic assembler functions **-** A simple SIC assembler - Assembler algorithm and data structures - Machine dependent assembler features **-** Instruction formats and addressing modes - Program relocation - Machine independent assembler features - Literals - Symbol-defining statements - Expressions - One pass assemblers and Multi pass assemblers - Implementation example - MASM assembler.  **UNIT III. LOADERS AND LINKERS**  Basic loader functions **-** Design of an Absolute Loader - A Simple Bootstrap Loader - Machine dependent loader features - Relocation - Program Linking - Algorithm and Data Structures for Linking Loader - Machine-independent loader features **-** Automatic Library Search - Loader Options - Loader design options - Linkage Editors - Dynamic Linking - Bootstrap Loaders - Implementation example - MSDOS linker.  **UNIT IV. MACRO PROCESSORS**  Basic macro processor functions - Macro Definition and Expansion - Macro Processor Algorithm and data structures - Machine-independent macro processor features - Concatenation of Macro Parameters - Generation of Unique Labels - Conditional Macro Expansion - Keyword Macro Parameters-Macro within Macro-Implementation example - MASM Macro Processor - ANSI C Macro language.  **UNIT V. SYSTEM SOFTWARE TOOLS**  Text editors - Overview of the Editing Process - User Interface - Editor Structure. - Interactive debugging systems **-** Debugging functions and capabilities - Relationship with other parts of the system - User-Interface Criteria. |

**Summary of Teaching Learning Methods**

The learning–teaching methodology will be student-centered with appropriate guidance of instructor/s during the students‘ activities .There will be Lecture, Demonstrations, Lab work Tutorials, Reading assignments and Group Discussions

**Summary of Assessment Methods**

The course will be assessed using the different assessment methods like: Quizzes, Reading assessments, Assignments, Project, Lab exam and Final exam

**References**

* Andrew Tanenbaum, Modern Operating Systems, 2nd Edition, USA, Prentice Hall, 2001
* William Stallings, operating Systems: Internals and Design Principles,5th Edition , Prentice-Hall,2005

**Course Outline #4**

|  |  |
| --- | --- |
| **Course Title** | Web design and Programming I |
| **Course Code** | SEng2074 |
| **CP** | 5 (2hr Lecture, 3hr Laboratory) |
| **Module Title** | Internet and Web Technology |
| **Module Number** | 07 |
| **Pre-requisites** | None |
| **Year** | II |
| **Semester** | II |
| **Status of Course** | Compulsory |

**Course Description**

Overview of the internet and World Wide Web; characteristics of web-based information systems; Client-server architecture; Web server and security, HTTP protocol; Web page design and development; information architecture & visualization; static & dynamic pages; HTML, CSS, client-side programming using scripting languages (JavaScript); introducing server-side programming (PHP) and web-based database application development.

**Learning Outcomes:**

At the end of this course the students will be able to:

* Practical use of Internet and its application in the real life
* Practical use of HTML and CSS.
* Practical use client-based programming
* Practical use of Server based programming

**Course Contents**

|  |  |
| --- | --- |
| **Week** | **Lecture Topics** |
| 1 | **CHAPTER 1 : OVERVIEW OF THE INTERNET AND WWW**   Computer program, programming language, Programming and the  Web, hypertext, www, HTML, Programming v. Scripting,  Client/Server Model , Universal Addressing (TCP/IP, DNS), Universal  Protocols (HTTP, URLs, HTML, FTP) |
| 2-4 | **CHAPTER 2 : WEB DESIGN AND DEVELOPMENT**  **FUNDAMENTALS**   HYPERTEXT MARKUP LANGUAGE (HTML): Why do we use  HTML?, HTML Document Structure, Basic HTML syntax, HTML  Tags |
| **5-6** | **CHAPTER 3:CASCADED STYLE SHEETS (CSS**)   Why use style sheets?, HTML vs. CSS, Anatomy of a CSS Rule,Linking HTML and CSS, |
| 7-8 | **CHAPTER 4: CLIENT-SIDE PROGRAMMING –JAVASCRIPT**   what is Client-side programming?, What is JavaScript?, Common  scripting tasks, limitations of client-side scripting, JavaScript and Java,  JavaScript Terminology(ObjCP, Properties, Methods, Events, Functions,  Values, Variables, Expressions, Operators, …) |
|  | **Chapter 5 :Advance topics client-side programming overview** |
| A9-11 | DOM and Interactivity |
| Asynchronous Data; ES6 |
| ReactJs |
| AngularJs |
| Firebase |
| Client-Side Routing (Thanksgiving) |
| Structuring React Apps |
| Addition client side programming framework |
| 12-15 | **CHAPTER 5: SERVER- SIDE PROGRAMMING-PHP**   Introduction to server-side programming – PHP: PHP Basic syntax, Send  Data to the Web Browser, Write Comments, Utilize Variables, Manipulate  Numbers, Work with constants, Flow Control, Manipulating Arrays,  String Manipulation, Working with Functions   Manipulating MySQL Databases with PHP: Connecting to Databases,  Send Data to a Database, Retrieve Data from a Database, Modify Existing  Data, Remove Existing Data |

**Summary of Teaching Learning Methods**

The course will be delivered in the form of lectures, demonstration, student presentations, group

Discussions, and individual and group project works.

**Assessment Methods:** - As per DMIT academic regulation

**References:**

1. Harvey M. Deitel. 2nd and 4 thed. Internet and World Wide Web: How to Program.

2. Goodman: Java script bible (3rd edition) Gage Publishing

3. Deitel and Deitel. "Java - How to Program", Addison-Wesley Press, Reading,

Mass.1998,

4. David Flanagan. "Java in a Nutshell (Java 1.1)", Second Edition, O'Reilly and

5. Associates Publishing, Sebastopol, CA, 1997.

6. David Flanagan. "Java Examples in a Nutshell (Java 1.1)", O'Reilly and Associates

Publishing, Sebastopol, CA, 1997.

7. Larry Wall and Randall Schariz. "Programming Perl", O'Reilly and Associates

Publishing. Sebastopol, CA, 1994

8. Scott Oaks and Henry Wong. "Java Threads", O'Reilly and Associates Publishing.

Sebastopol, CA, 1997.

9. Gary Cornell, Cay Horstmann. "Core Java", SUN Soft Press Publishing, Mountain View, 1996.

10. S. Gundavaram. "CGI Programming on the World Wide Web", O'Reilly and

Associates Publishing, Sebastopol, CA, 1996.

11. http://www.w3schools.com- the best website to learn web development

12. Internet is your main resource.

**Course Outline #5**

|  |  |
| --- | --- |
| **Course Title** | Object Oriented Programming |
| **Course Code** | SEng2062 |
| **CP** | 5 (2hrs Lecture, 3hrs Laboratory) |
| **Module Title** | Advanced Programming |
| **Module Number** | 07 |
| **Pre-requisites** | Fundamentals of Programming II (SEng2021) |
| **Year** | II |
| **Semester** | II |
| **Status of Course** | Compulsory |

**Course Description**

Student’s previous exposure to and skills in structured programming is assumed (although a review of the necessary basic programming notions is included). The course is designed to introduce how to develop business applications using object-oriented design methodology with Java as an illustration programming language. It includes Object Oriented Programming paradigm and its use; classes, ObjCP, Abstraction and Encapsulation; Inheritance, Polymorphism, Creating Graphical User interfaces (GUIs), Data Structures, Exceptions(Try, catch, and throw, finally how exceptions affect the design of an application), File input/output (I/O), Threads and Java Database Connectivity

**Learning Outcomes**

At the end of the course students will be able to:

* understand the principles of software engineering with emphasis on the various phases of the software development life cycle
* Understand major concepts of object-oriented programming
* understand the programming environment as defined by compilers, interpreters, editors, and other system software providing support for the programming activity
* develop skills in OO design and program development within an integrated development environment
* Use arrays and other data structures
* To understand the concepts of encapsulation, inheritance
* Implement I/O functionality to read from and write to data and text files.
* understand object technology and its applications
* Explain the application of a variety of data structures, understand the advantages and disadvantages of those structures
* To be familiar with object-oriented design concepts, and have had practical experience of designing and implementing object-oriented software

**Course contents**

|  |  |
| --- | --- |
| **Weeks** | **Lecture Topics** |
| **1-2** | **1. Chapter 1: Introduction to Object-Oriented Programming (OOP)**  1.1. Overview of OOP?  1.2. Why Java?  1.3. The JVM and Byte Code  1.4. Basic concepts of OOP  1.4.1. classes  1.4.2. objects  1.4.3. members  1.4.4. class member visibility  1.4.5. encapsulation, inheritance and polymorphism |
| **3-5** | **2. Chapter 2: The inside of objects and classes: More on OOP concepts**  2.1. member methods and their components  2.2. instantiation and initializing class objects  2.3. constructors  2.3.1. default and parameterized  2.3.2. overloaded constructors  2.4. methods  2.5. access specifiers  2.6. accessors and mutators  2.7. calling and returning methods  2.8. static and instance members |
| **6-9** | **3. Chapter 3: Inheritance**  3.1. Concept of inheritance  3.2. Superclasses and subclasses  3.3. Protected members  3.4. Overriding methods  3.5. Using this() and super()  3.6. Use of final with inheritance  3.7. Constructors in subclasses |
| **10-12** | **4. Chapter 4: Polymorphism**  4.1. Introduction  4.2. Relationships among objCP in an inheritance hierarchy  4.3. Assigning reference of subclass to superclass-type variable  4.4. Assigning a superclass reference to subclass-type variable  4.5. Subclass method calls via superclass-type variable  4.6. Summary of allowed assignments between superclass and subclass variables  4.7. Multiple inheritance and interfaces |
| **13-14** | **5. Chapter 5: Exception Handling**  5.1. Exception handling overview  5.2. The causes of exceptions  5.3. The Throwable class hierarchy  5.4. Handling of an exception  5.5. The throw statement  5.6. The finally clause  5.7. User defined exceptions |
| **15-16** | **6. Chapter 6: Files and Streams**  6.1. Introduction  6.2. I/O classes  6.3. File and FileDialog objects  6.4. Low-Level File I/O  6.5. High-Level File I/O  6.6. Object I/O  6.7. Random Access files |

**Summary of Teaching Learning Methods**

The teaching-learning methodology will be student-centered with appropriate guidance of instructor/s during the student’s activities. The course will be delivered in the form of lectures,lab session, group discussions, and individual and group project works.

**Assessment Methods**: - As per DMiT academic regulation

**References**

1. Y. Daniel Liang, 2009. Int. to Java Programming Comphrensive version, Seventh Edition. Atlantic State University

2. Deitel, 2010, Java for Programmers Second edition. Prentice Hall

3. Deitel, 2004. JAVA how to program.5th Ed. New Delhi: Prentice-Hall of India

4. R. Sahoo& G. Sahoo, 2001. JAVA 2 The complete reference book; New Delhi: Prentice- Hall of India

5. C. Thomas Wu, 1999. An introduction to Object-Oriented Programming with Java; McGraw Hill.

6. John Lewis & William Loftus, 1998. JAVA Software Solutions: - Foundations of Program Design; Addison-Wesley.

**Course Outline #6**

|  |  |
| --- | --- |
| **Course Title** | **Fundamentals of Networking** |
| **Course Code** | SEng2072 |
| **CP** | 5 (2hrs Lecture, 3hrs Laboratory) |
| **Module Title** | Computer Networking and Security |
| **Module Number** | 07 |
| **Pre-requisites** | None |
| **Year** | III |
| **Semester** | I |
| **Status of Course** | Compulsory |

**Course Description**

The course aims at exploring the various types of data communication systems, networks and their applications. The content includes: computer networks, seven-layer architecture, OSI & TCP/IP suite of protocols, network hardware, network software, standardization, guided transmission media, wireless transmission, data link layer, Ethernet and IP addressing. It involves practical session on Cabling and crimping, Configuring TCP/IP, Peer to Peer Networking, Sharing Files, Sharing Printers, Client-server Networking, Steps for Creating a home or small office Network, Experiencing collaboration tools, installing & Configuring Network Operating System, Exploring Server Roles, Setting up a DNS Server, setting up a DHCP server, Domain controller and IP Addressing.

**Learning Outcomes**

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

* Describe the basics of data communications and network
* Explain the benefits and the need for network
* Understand data transmission and transmission media
* Understand Protocols and various networking components
* Understand TCP/IP & OSI Reference Model
* Demonstrate cable crimping, establishing, setup and troubleshooting Networks
* Demonstrate network addressing
* Understanding network Equipment
* Understand network security and data integrity

**Course contents**

|  |  |
| --- | --- |
| **Duration(**  **Week)** | **Topic** |
| **1** | **Chapter 1: Introduction**  1.1. History & overview of Networks  1.2. The impact of Networks on daily life  1.3. The network as a platform  1.4. Network Role & Elements  1.5. Network Architecture Characteristics  1.6. Computer Networks Versus Human Network |
| **2** | **Chapter 2: Data Communications**  2.1. What is communication?  2.2. The platform for communication  2.2.1. Communicating the Message  2.3. Data transmission  2.3.1. Concepts and Terminology  2.3.2. Analog and Digital Data Transmission  2.3.3. Transmission Impairments  2.4. Components of the network  2.4.1. End Devices & their role  2.4.2. Intermediary Devices & their role  2.4.3. Network Media |
| **3** | **Chapter 3: Network Types**  3.1. LANs, WANs and Internetworks  3.2. Peer to peer versus Server based Networks  3.3. Packet-switched and Circuit switched networks  3.4. Network cabling & Topologies |
| **4** | **Chapter 4: Protocols**  4.1. Rules & Network Protocols  4.2. Protocol suites & Industry Standards  4.3. Layered Models  4.3.1. The TCP/IP Model  4.3.2. The OSI Model  4.3.3. Comparing OSI Model with TCP/IP Model  4.3.4. Overview of familiar Protocols |
| **5** | **Chapter 5: OSI Reference Model**  5.1. Layered Framework of OSI  5.2. Overview & functions of each layer  5.2.1. BiTs, Datagram, packet & Frames  5.2.2. Physical Signaling & Encoding |
| **6** | **Chapter 6: Switching & Multiplexing**  6.1. Switching Concept and Types  6.2. Multiplexing Concepts and Types  6.3. **Introduction to Ethernet & Wireless Networks**  6.3.1. **Ethernet and Fast Ethernet**  6.3.2. **Introduction to Wireless Network** |
| **7** | **Chapter 7: Introduction to IP Addressing and Subnetting**  7.1.**Classful& Classless Addressing**  7.2.**Subnetting and Variable Length Subnet Masking(VLSM)** |
| **8** | **Chapter 8: Data Security and Integrity**  8.1. Fundamentals of secure networks; cryptography  8.2. Encryption and privacy  8.3. Authentication protocols  8.4. Firewalls  8.5. Virtual private networks  8.6. Transport layer security |
| **Device configuration part** | **Add here** |

**Summary of Teaching Learning Methods**

The course will be delivered in the form of lectures, demonstration, student presentations, group

Discussions, and individual and group project works.

**Assessment Methods:** - as per DMIT academic regulation

**References**

1. Data Communications and Networking, 4th Ed., Behrouz A. Forouzan

2. A S Tannenbaum" Computer Networks" Prentice Hall of India Publication, 2002

3. Data and Computer Communications, 8th ed., William Stallings

4. Computer Networking. Kurose & Ross. Addison Wesley

5. Fred Halshall "Data Communication, Computer Networks & Open systems" Publication

Pearson Education

6. Any Cisco Material (CCNA Module1) will be helpful