Computer Science and Technology
Draft Syllabus (4th Semester)

Microprocessor & Programming

Name of the Course: Microprocessor & Programming	
Course Code: MP	**Semester: Fourth**
Duration:16 weeks	**Maximum Marks: 100 (Theory) + 50 (practical)**
Teaching Scheme	**Examination Scheme**
Theory: 3 hrs./week	Mid Semester Exam.: 20 Marks
Tutorial: hrs./week	Assignment & Quiz: 10(Th.)+25(Pr) Marks
Practical: 2 hrs./week	End Semester Exam.: 70(Th)+25(Pr) Marks
Credit: 3+1	

Aim: To understand Digital electronics and able to design digital circuit and to understand A/D and D/A converter

Sl. No.	
1.	To study Architecture and memory management of 8 bit & 16 bit microprocessor (i.e. 8085 & 8086).
2.	To study assembly language programming technique and use of DEBUG command.
3.	To implement different system interfacing.

Objective: Student will be able to

Sl. No.	
1.	Draw block diagram for architecture of 8085 and to know all the pin function.
2.	Draw block diagram for architecture of 8086 and to know all its pin function.
3.	Describe concepts of pipelining segmentation and address generation.
4.	To know the instruction set and addressing modes.
5.	Write the efficient Assembly Language Program for different problem statements and use of procedures and macros.
6.	Design interface of memory chips.
7.	Design and verify Sequential circuit.

Pre-Requisite:

Sl. No.	
1.	Basic knowledge computer architecture and digital electronics is helpful.
2.	

Contents (Theory)		Hrs./Unit	Marks
Unit: 1	**Basics of Microprocessor** 1.1 Evolution of Microprocessor and types 1.2 Silent features of 8085 Microprocessor, architecture of 8085 (Block diagram), pin diagram, register organization, limitations of 8-bit Microprocessor. 1.3 8085 interrupt structure	6	
Unit: 2	**16-bit Microprocessor 8086** 2.1 Silent features of 8086 Microprocessor, architecture of 8086 (Block diagram, signal description), register organization, concepts of pipelining, 2.2 memory segmentation and memory address	10	

	generation from segment offset address. 2.3 Minimum and Maximum Mode operation and diagram 2.4 8086 interrupt structure.		
Unit: 3	**8086 Instruction set** 3.1 Concept of Machine Language, Instruction format, addressing modes. 3.2 Instruction set (Arithmetic, logical, data transfer, bit manipulation, string, program control transfer, process control)	06	
Unit: 4	**The art of assembly Language Programming** 4.1 Assembly Language Programming Tools Editors, Assembler, Linker, Debugger. 4.2 Assembler directives, model of 8086 assembly language programming, programming using assembler.	06	
Unit: 5	**Procedure and Macro** 5.1 Defining Procedure (Directives used, FAR and NEAR, CALL and RET instructions) 5.2 Defining Macros. 5.3 Assembly Language Programs using Procedure and Macros. 5.4 DOS interrupt services.	08	
Unit: 6	**System Interfacing** 6.1 Interfacing Techniques (I/O mapped I/O, Memory mapped I/O, memory and I/O addressing, 8086 addressing, and address decoding, memory interfacing as Even and Odd bank) 6.2 Interfacing 8255, Block diagram, modes of operation. 6.3 8259: Block diagram, Characteristics and function only. 6.4 8257/8237: Block diagram, Characteristics and function only.	09	
Total		**45**	

Practical:

Skills to be developed:

Intellectual skills

- Use of programming language constructs in program implementation.
- To be able to apply different logics to solve given problem.
- To be able to write program using different implementations for the same problem
- Study different types of errors as syntax semantic, fatal, linker & logical
- Debugging of programs
- Understanding different steps to develop program such as

  ☐Problem definition

  ☐Analysis

  ☐Design of logic

  ☐Coding

  ☐Testing

  ☐Maintenance (Modifications, error corrections, making changes etc.)
- Motor skills
- Proper handling of Computer System**.**

List of Practical:

1) Basics of Assembler, linker, debugger, editor
2) Write an Assembly Language Program to
   - Add / Sub two 16 bit numbers.
   - Find sum of series of numbers.
   - Multiply two 16 bit unsigned/ signed numbers.
   - Divide two unsigned/ signed numbers (32/16 , 16/8, 16/16, 8/8)
   - Add / Sub / Multiply / Divide two BCD numbers.
   - Find smallest/ largest number from array of n numbers.
   - Arrange numbers in array in ascending/ descending order.
   - Perform block transfer data using string instructions / without using string instructions.
   - Compare two strings using string instructions / without using string instructions.
   - Display string in reverse order, string length, Concatenation of two strings.
   - Convert Hex to Decimal, Decimal to Hex.

   ** Practical can also be done by using DEBUG command. Any program other than those given in the list will be appreciated.

Text Books:

Name of Authors	Title of the Book	Edition	Name of the Publisher
Krishna Kant	Microprocessors and Microcontrollers		PHI
Ray &Bhurchandi	Advance Microprocessor and Peripherals		TMH
Hall	Microprocessors and Interfacing		TMH
Kumar, Saravanan&Jeevanantham	Microprocessor and Microcontroller		Oxford
Savaliya	8086 Programming and advance processor architecture		Willy

Reference Books:

Name of Authors	Title of the Book	Edition	Name of the Publisher
Chhabra	The Intel 8086/8088 microprocessor Architecture, Programming Design & Interfacing		DhanpatRai

Suggested list of Assignments / Tutorial:

Sl. No.	Topic on which tutorial is to be conducted (To be given as per Lab experiment list)
1.	

Computer Network

Name of the Course: Computer Engineering Group (**Computer Network**)	
Course Code:	**Semester: FOURTH**
Duration:	**Maximum Marks: 150 (Practical 25+25)**
Teaching Scheme	**Examination Scheme**
Theory: 3 hrs./week	Class Test: 20 Marks
Tutorial: hrs./week	Teachers Assessment: 10 Marks
Practical: 2 hrs./week	End Semester Exam.: 70 Marks
Credit: 4	

Aim:	
Sl. No.	
1.	To learn basic concepts of Computer networks.
2.	To study hardware in detail required for networking.
3.	To learn in detail basic models of networking -ISO OSI and TCP/IP.
Objective:	
Sl. No.	Students will able to:
1.	☐ Identifying the benefits of network.
2.	☐Distinguish between Network classifications.
3.	☐ Describe different types of Topology.
4.	☐Describe different types of Network devices.
5.	☐Compare different transmission media.
6.	☐Compare OSI and TCP/IP model.
7.	☐ Configure TCP/IP.

Pre-Requisite:	
Sl. No.	
1.	Fundamentals of Programming Languages

Contents (Theory)		Hrs./Unit	Marks
Unit: 1	**INTRODUCTION TO DATA COMMUNICATION NETWORKING** 1.1 Data communications: components, data representation. 1.2 BASIC CONCEPTS**:** Servers, Client, Workstation, Hosts (definition & applications) 1.3 TYPES OF COMPUTER NETWORKS**:** LAN, MAN and WAN. 1.4 TYPES NETWORK ARCHITECTURE**:** Peer-to-peer, Client-Server and Distributed. 1.5 Simplex, Half duplex and Full duplex 1.6 Parallel and Serial, Asynchronous and Synchronous 1.7 Definition and different types of Noise, Nyquist rate, Shannon's Capacity. 1.8 Network Features - File Sharing; Printer Sharing; Application Services; EMail; Remote Access.	**05**	
Unit: 2	**Network Topologies and Networking Devices:** 2.1 Type of Topology - Bus Topology; Ring Topology; Star Topology; Mesh Topology; Tree Topology; Hybrid Topology. 2.2 Network Control Devices -Hubs; Switches; Routers; Bridges; Repeaters; Gateways; Modems	**03**	
Unit: 3	**Transmission Media:** 3.1 Guided Media -Twisted Pair -UPT, STP; Coaxial Cable; Optical Fiber - Optical Fiber Structure, Light Source for Fiber, Propagation Mode, Advantages of optical fiber and Disadvantages of optical fiber. 3.2 Un-Guided Media: Wireless Communication – Communication Band; Microwave Communication; Satellite Communication – Access Method; Cellular (Mobile) Telephone – Band in Cellular	**04**	

	Telephony, Calls Using Mobile Phones, Transmitting receiving operations; New Developments.		
Unit: 4	4.1 OSI Reference Model - Interlayer Communication – Data Encapsulation, Horizontal Communication, Vertical Communication, Encapsulation Terminology; Physical layer; Data link layer; Network layer; Transport layer; Session layer; Presentation layer; Application layer. 4.2 TCP/IP Reference Model – Link; Internet; Transport; Application layer. 4.3 Comparison of the OSI and TCP/IP reference models.	**03**	
Unit: 5	5.1 MUL**TIPLEXING:** FDM, TDM, WDM, ADM, OFDM. 5.2 SWITCHING**:** Circuit Switching : time division & space division switch, Packet Switching, Message Switching.	04	
Unit: 6	**Data link layer** 6.1 Types of Error, Framing(character and bit stuffing), error detection & correction methods. 6.2 Flow control and Error control mechanism.	04	
Unit: 7	**Medium access sub layer** 7.1 Point to point protocol, FDDI, token bus, token ring; Reservation, polling. 7.2 Medium Access Control: Motivation for a specialized MAC: Hidden and Exposed terminals. Near and Far terminals; **7.3** FDMA, TDMA: Fixed TDM, Classical Aloha, Slotted Aloha, Carrier sense multiple access, Demand assigned multiple access, Multiple access with collision detect, Multiple access with collision avoidance, Inhibit sense multiple access; CDMA;	**05**	
Unit: 8	8.1 Protocols, Services and Standards (in brief): X.25, ATM, ISDN, Token Ring and Token Bus.	**02**	
Unit: 9	**9.1 Routing** : techniques, static vs. dynamic routing , routing table for classful address; Routing algorithms: shortest path algorithm, flooding, distance vector routing, link state routing; 9.2 IP Addressing - IP Address Assignments; IP Address Classes; Subnet Masking; Registered and unregistered Addresses. Congestion control algorithm: Leaky bucket algorithm, Token bucket algorithm, choke packets; 9.3 Quality of service: techniques to improve Qos.	**04**	
Unit: 10	**TCP/IP Fundamentals:** 10.1 TCP/IP Protocols - SLIP and PPP; ARP; IP; ICMP; TCP and UDP.	03	
Unit: 11	**APPLICATION LAYER** Definition of Internet and compare with Intranet – URL – HTTP – HTML. DNS; SMTP, SNMP, FTP, WWW;	03	
Unit: 12	**NETWORK SECURITY** 2.1 Encryption (Private and Public key) – Decryption – Digital Signature. 12.2 Firewalls **Cyber Security**	05	

	12.3 Introduction to Cybercrime: Definition- Cybercrime and Information Security – Classification of Cybercrimes. 12.4 **Cyber offenses**: Introduction- Criminals Plan the Attacks – Social Engineering – Cyber stalking – Attack Vector – Cloud Computing		
Total		**45**	

Contents (Practical)	
Sl. No.	Skills to be developed
1.	**Practical:** Skills to be developed: Intellectual skills: ☐Fault finding of network ☐Troubleshooting of network ☐Proper installation of network Motor skills: ☐Proper handling of Computer System hardware.☐Testing ☐ Maintenance (Modifications, error corrections, making changes etc.)
2.	Motor Skills: ☐ Proper handling of Computer System.

List of Practical:

LIST OF SAMPLE PROBLEMS FOR DATA STRUCTURE LAB(for example)

1 Creating Windows 2003 Server Boot Disk.
2 Installing Windows 2003 Server &UNIX / Linux
3 Installing Active Directory
4 Creating AD Objects
5 Setting up Local Print Device & Installing and Configuring a Network – Capable Print Device
6. Create new Users & give the Permission
7 Use step by step procedure for i.e. File sharing & Printer sharing.
8 Compare different Network Topologies.
9 Compare Network directing devices.
i.e. Hub, Switch, Router.

10 To study crimping: RJ-45, RJ-11, Cross-over Cable and Create a Network cable using RJ45 connectors.

11. To study the different expansion slots of a motherboard, set the NIC to expansion slot and to install the driver.
12 To locate MAC address of computer.

13. To make a peer-to-peer Network System.
14. Implementing a TCP/IP Network configuring

15. To run the following application in a network system and get knowledge**:**

(i) FTP, (ii) Telnet, (iii) Mail, and, (iv) Talk.

16. To use the ping utility in order to understand its use in a troubleshooting environment.

17. To be familiar with loop back testing.

18. To be familiar with the idea of socket and to write a socket program.

_____

Text Books:			
Name of Authors	Title of the Book	Edition	Name of the Publisher
Prakash C. Gupta	**Data Communications and computer Networks**	2nd	PHI
DR. Sanjay Sharma	**A Course in Computer network**		KATARIA
N. Olifer, V. Olifer	**Computer Networks Principles, Technologies and protocols for network Design**		WILEY
Uyless Black	Computer Networks Protocols, Standards, and interface		PHI
Nina Godbole&SunitBelapure	**CYBER SECURITY**		WILEY India
Reference Books:			
Name of Authors	Title of the Book	Edition	Name of the Publisher
A.S.Tanenbaum	Computer networks		PHI
B.A.Farouzan	Data communication and networking		TATA McGraw hill

Suggested list of Laboratory Experiments:	
Sl. No.	Laboratory Experiments
1.	Basic TCP/IP utilities and commands. (eg: ping, ifconfig, tracert, arp, tcpdump, whois, host, netsat, nslookup, ftp, telnet etc...)
2.	Configure a router (Ethernet & Serial Interface) using router commands including access lists on any network simulator (eg. packet Tracer)
3.	Network design and implementation for small network using actual physical components with IP address scheme

Suggested list of Assignments / Tutorial:	
Sl. No.	Topic on which tutorial is to be conducted
1.	Configuration of any three of the following of for each student a) Remote Login Service – TELNET/SSH b) Configuration of FTP server and accessing it via FTP Client.
2.	Installation of NS-2. Test network animation on Network Simulator2 (NS2).

Relational Database Management System

Name of the Course:Relational Database Management System	
Course Code: RDBMS	**Semester: Fourth**
Duration:	**Maximum Marks:100(Theory) + 100 (practical)**
Teaching Scheme	**Examination Scheme**
Theory: 3 hrs./week	Mid Semester Exam.: 20 Marks
Tutorial: hrs./week	Assignment & Quiz: 10(Th.)+50(Internal Practical) Marks
Practical: 3hrs./week	End Semester Exam.: 70(Th)+50(External Practical)Marks
Credit: 3+1	

Aim:	
Sl. No.	
1.	To study and understand the basic concepts of RDBMS.
2.	To learn SQL and PLSQL in detail.
3.	To learn how to work with any database.
Objective: Student will be able to	
Sl. No.	
1.	Understand the concept of Database system and Client Server Architecture
2.	Understand and develop the concepts of Data Modeling, Security and Integrity.
3.	Understand and execute different SQL queries and PL / SQL programs.
4.	Normalize the database using normal forms.
5.	Understand the concept of query processing and Transaction processing.

Pre-Requisite: Basic knowledge of computer is helpful.			
Contents (Theory)		Hrs./Unit	Marks
Unit:1	**Database System Concept & 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 1.4 Client Server Architecture:	10	
Unit: 2	**Relational Data Model and Security and Integrity Specification** 2.1 Relational Model: Basic concepts, attributes and domains, Keys concept : Candidate 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.	8	
Unit: 3	**SQL and PL-SQL** 3.1 Introduction to SQL queries,Creating ,Inserting ,Updating and deletingtables and using constraints, Set operations & operators, Aggregate functions ,string functionsand date ,time functions, Null values, Nested sub queries, Complexqueries,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. Error handling in PL/ SQL	14	
Unit: 4	**Relational Database Design, Storage and File systems**. 4.1 Purpose of Normalization, Data redundancy and	8	

	updating anomalies, Functional Dependencies and Decomposition, 4.2 Process of Normalization using 1NF, 2NF, 3NF, multivalued dependencies and BCNF. 4.3 E-R Model details. 4.4 File Organization, Organization of records in files, Storage of Object Oriented databases, Basic concept of Indexing and Hashing.		
Unit: 5	**Query Processing and Transaction Processing** 5.1 General strategies for query processing, Equivalence expressions, Selection & join operation. 5.2 Concept of transaction, States of transactions, Concurrent Executions, Serializability Recoverability, Transaction Definition in SQL.	5	
Total		**45**	

Contents (Practical)

Skills to be developed:
Intellectual skills:
1. Develop the fields of data base
2. Decide proper specifications
3. Query Processing and transaction processing

Motor skills:
1. Prepare appropriate data tables
2. Sequential writing of steps

List of Practical:

1) Creating & Executing DDL in SQL.
2) Creating & Executing Integrity constraints in SQL.
3) Creating & 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 50 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.

**** Practice of different types of Query is essential. Use of any "open source database software" is highly appreciated.**

Suggested List of Laboratory Experiments :

1 VB database connectivity
2 Miniproject-1
3 Miniproject-2

Suggested List of Assignments/Tutorial :

1 Create ER diagram for student database.
2 Create ER diagram for Hospital management.
2 Write difference between DDL and DML.

Text Books:			
Name of Authors	Title of the Book	Edition	Name of the Publisher
Korth	Database Sytem Concept		TMH
C J Date	An Introduction to Database System		Pearson
Navathe	Fundamentals of Database System		Pearson
2006 ISRD Group	Introduction to Database Management System		TMH
Desai	An Introduction to Database System		West publishing Company
Allen	Introduction to Relational Databases and SQL programming.		Wiley
Raghu Ramakrishnan, Johannes Gehrke	Database Management Systems		TMH
Reference Books:			
Name of Authors	Title of the Book	Edition	Name of the Publisher
Deshpande	SQL and PL/SQL for Oracle 11g		Dreamtech

Object oriented Programming

Name of the Course : Object Oriented Programming	
Course Code:	**Semester: Second**
Duration: Six Months	**Maximum Marks: 150**
Teaching Scheme:	**Examination Scheme:**
Theory: 3 Hrs/week	Class Test: 20 Marks, TA: 10
Tutorial: Nil	Assignment & Sessional: 25 (Internal)+25 (Ext.)
Practical/ Sessional: 3 Hrs/week	End semester Exam: 70
Credit: 3 + 1	
Aim of the Course:	
S. No	Aims about
1.	The aim of this course is to teach the principles underlying Object Oriented Programming through C++
2.	To increase reusability in programming.
3.	To reduce the costs of developing and adapting software to meet new requirement.
Objective of the course:	
S. No	The students will be able to -
1.	Write programs using objects & classes.
2.	Develop programs to create and destroy the objects using constructors and Destructors.
3.	Use existing operators for different meanings in Operator Overloading concept.
4.	Using reusability concept through Inheritance concept.
5.	Implement pointers for arrays, strings & object.
6.	Describe polymorphism, concepts, its types, virtual function & write program for same.
7.	Apply formatted & unformatted console I/O operation & perform file related activities using C++ streams.
Pre-Requisites -	
S. No	
1.	Interaction with DOS / Windows Operating System.
2.	Ability to develop logic / flow of simple problem.
3.	Basic Concepts of 'C'.

Unit No.	Contents	Hrs/Unit	Marks
1	**Concept of Object Oriented Programming.** 1.1 **History & features:**It's need & requirement, procedure oriented programming versus object oriented programming, basic concepts object oriented programming, object oriented languages, object based languages. 1.2 **Beginning with C++**: Concepts & structure of C++ programming, insertion and extraction operators, objects of input and output stream class. Uses of iostream.h header file.	5	
2	**Objects & Classes:** 2.1 Specifying a class, Defining member functions, Arrays within a class, Creating objects, memory allocation for objects, static data & member function, Arrays of objects, objects as function argument. 2.2 Class specifiers and their uses, distinction between structure (struct) of C and Class.	5	
3	**Constructors and Destructors**. 3.1. Concept of Constructor (Default, Parameterized, Copy), Zero argument and explicit Overloaded Constructors, Destructors and properties, uses of destructors. **Function and Operator Overloading** 3.2 Function overloading, Inline member functions, constant member functions. 3.3 Operator overloading (overloading unary & binary operators), rules for overloading operators. Type Conversion: Conversions from basic to class type, class to basic type, class to class type. Operators that can not be overloaded.	6	
4	**Inheritance** 4.1. Concepts of inheritance, Derived classes, Member declaration (Protected), Types of inheritance (Single, multilevel, multiple, hierarchical, Hybrid inheritance), Ambiguity in multiple inheritance. 4.2 Virtual base classes, Abstract classes, Constructors in derived classes. 4.3 Class within class, containership, IS A and HAS A relationship and their differences, Namespaces. 4.4 Friend function, Friend Class, advantages and disadvantages of friends.	6	
5	**Pointers in C++** 5.1. Concepts of pointer (Pointer declaration, pointer operator, address operator, pointer expressions, and pointer arithmetic), Pointers & functions (Call by value, call by reference. 5.2. Pointers & objects (Pointers to objects, this pointer, and pointer to derived classes). 5.3. Memory management through pointer: new, delete, operators and free(), malloc(), calloc() functions, Member dereferencing Operators.	8	
6	**Polymorphism** 6.1. Concepts of polymorphism, types of polymorphism, Overloading & overriding, Overloading Virtual function, Static & dynamic binding. 6.2 Pure Virtual functions, Virtual Constructors and Destructors.	5	

7	**Exception Handling** Concepts and uses of exception handler, the try /throw/ catch construct, uses and implementation of multiple exceptions, limitation of exception handling.	**4**	
8	**Templates** Concepts of Templates, Function and Class Templates, Advantages of templates.	**2**	
9	**Basic function of I/O system basics & File Processing** Stream classes, using formatted & unformatted functions, using manipulator to format I/O, Basics of file system, opening & closing a file, reading & writing character from a file (get, put, get line, write), Command line arguments.	**5**	

Practical / Sessional Works

Skills to be developed**:**

Intellectual skills:

- Use of programming language constructs in program implementation.
- Apply different logics to solve given problem.
- Write program using different implementations for the same problem.
- Identify different types of errors as syntax, semantic, fatal, linker & logical.
- Debugging of programs.
- Understanding different steps and stages to develop complex program.

Motor Skills:

- Proper handling of Computer System**.**

A sample List of Practical / Sessional works to be done (Leading '*' denotes the harder problems)

S. No.	Specific problem(s) related with practical / sessional work	Skill area
01	i) Programs to input & output data (Simple programs). ii) Write a program which read a value and print to decimal, octal and hexadecimal. iii) *Displaying entered number with different manipulators like setbase, setw, setprecision etc.	Formatted output. (Any two)
02	iv) To create a simple class with three different member data (int, float and char). Write member function to insert data into those members and display them accordingly. v) To find greatest / smallest of three numbers using OOP in C++. vi) Create a student class with data members as roll, name and marks with respective data types as int, chars and float. Now create n objects of student type and insert data into those objects. Display the student information who got the highest mark. vii) Write an OOP in C++ to add, subtract and multiplication of two matrices of size 3X3. viii) Create a class complex with real and imaginary part (integer). Implement default, parameterized and copy constructor to initialize the objects of complex class and display them. ix) Implement Destructors. x) *Create a class complex as above. Now add, subtract and multiply on two objects of complex type i) using objects as function argument, ii) returning object from function. xi) *Create a class distance with foot and inch. Now add and subtract between two objects of distance type i) using objects as function argument, ii) returning object from function. xii) Implement a counter class with a static member count. Create different objects of counter class to show the behaviour of count.	Class, object, arrays of objects, member data & member function.
03	*Design a base class which has following data members with requisite data types. a) Name, b) Roll, c) Phnno, d) Address. Then design a derived class from above base	Inheritance

	class with member data as a) marks1, b) marks2, c) total (should not be inserted). Now display the result of n student consisting roll, name, total. Show ambiguity in inheritance and implement the method to avoid it. Implement containership. *Implement constructor inheritance.	
S. No.	Specific problem(s) related with practical / Sessional work	Skill area
04	xvii) Write a program which reads a complex number. Now increment only the real part and display the same. xviii) Write down a program which reads a complex number. Now decrement the real and imaginary part and display. xix) Implement both prefix and postfix operation on a complex number. xx) Overload arithmetical binary operators (+, -, *) for complex numbers. xxi) *Overload comparison operators (<, >, <=, >=, !=, ==) for two objects of same type. xxii) Write a program which converts one basic type to class type. xxiii) *Write a program which converts one class type to another class type. xxiv) *Implement friend function to access the data members from two different classes.	Operator and function overloading
05	xxv) *Write a program in C++ using pointer which calculate the sum of two complex numbers. xxvi) *Write a program to create a matrix using pointer in dynamic way (pointer to an array and array of pointers). xxvii) Uses of **this** pointer to access the content of an object.	Pointers
06	xxviii) Implement Compile time Polymorphism (early bindings) and run time Polymorphism (late bindings) using virtual function. xxix) Implement friend class using forward declaration to access the private data member of the other.	Polymorphism
07	xxx) Write a program which generates a template class, by which we can perform integer type data addition and float type data addition also. xxxi) *Use of function template with multiple parameters. xxxii) *Use of class template with multiple parameters. xxxiii) Write a program for division operation to handle an exception if the divisor is 0. xxxiv) *Write a program in C++ to handle multiple exceptions for different operational output.	Templates & exception Handling
08	xxxv) Use different modes of opening files to perform various operations on file. xxxvi) *Create a random file to insert, edit and delete operations using file pointers and manipulators. xxxvii) Write a program for reading and writing objects into a file.	I/O Operations on files through Stream

Text Books

Name of the Authors	Titles of the Book	Edition	Name of the Publisher
SouravSahay	Object Oriented Programming with C++	Second Edition	Oxford
Robert Lafore	Object Oriented Programming in C++	Fourth Edition	Pearson
B Stroustrup	C++ programming Language	3rd Edition	Pearson
Bhushan Trivedi	Programming with Ansi C++	Second Edition	Oxford
M.T. Somashekhara, D.S. Guru, H.S. Nagendraswamy, K.S. Manjunatha	Object Oriented Programming with C++		PHI
E. Balgurusamy	Object oriented programming with C++		Tata McGraw Hill
Sunil K Pandey	Thinking in C++	Seventh Edition	S. K. Kataria and Sons

Websites:

- http://www.sourcecodesworld.com
- http://www.softeam.com
- http://www.cplus.about.com/od/beginnerctutorial

Demo lectures with power point presentations using LCD projector should be arranged to develop Programming concepts of students.

Computer Graphics

Name of the Course: Computer Graphics	
Course Code: CG	**Semester: Fourth**
Duration:16 weeks	**Maximum Marks: 100 (Theory) + 50 (practical)**
Teaching Scheme	**Examination Scheme**
Theory: 3 hrs./week	Mid Semester Exam.: 20 Marks
Tutorial: hrs./week	Assignment & Quiz: 10(Th.)+25(Pr) Marks
Practical: 2 hrs./week	End Semester Exam.: 70(Th)+25(Pr) Marks
Credit: 3+1	

Aim: To understand different aspects of computer graphics and use.	
Sl. No.	
1.	The chief aim of computer graphics is to display and print realistic-looking images
2.	Understand the principles of 3D computer graphics
3.	Develop programming skills for computer graphics Programming in C.

Objective: Student will be able to	
Sl. No.	
1.	To apply the algorithms to draw lines, circles and polygons.
2.	To use transformation techniques to scale, rotate and translate the object.
3.	To select the methods of enlarging visible portion of drawing.
4.	To develop the logic for drawing the natural objects using different algorithms for curved lines.
5.	To describe the fundamentals of raster graphics and interactive graphics.

Pre-Requisite:	
Sl. No.	
1.	Basic knowledge of C programming
2.	Basic data structure.
3.	Concept of mathematics.(Geometry, Matrix and other field).

Contents (Theory)		Hrs./Unit	Marks
Unit: 1	**Basics of Computer Graphics** 1.1 Display devices, Primitive operations, 1.2 Text mode and graphics mode, graphics functions, Shapes, colors, Co-ordinate systems, 1.3 Applications of computer graphics 1.4 Raster scan display, Random scan display	6	
Unit: 2	**Line, circle, and polygon.** 2.1 Basic concepts in line drawing, 2.2 Line drawing algorithms: DDAalgorithms, Bresenham's algorithm Circle generating algo algorithm, 2.3 Bresenham's circle drawing algorithm, midpoint circle drawing algorithm. 2.4 Polygons – Types of polygons, Polygon representation, inside –outside test, 2.5 Polygon filling: Flood fill, scan-line algorithm.	13	

Unit: 3	**Transformations** 3.1 2D transformation: Translation, Rotation, scaling, Reflection, shearing, transformation matrices, Homogeneous co-ordinate system. 3.2 Rotation about an arbitrary point, scaling about fixed point. 3.3 Composite transformations. 3.4 3D Transformation: scaling, rotation, translation, rotation about arbitrary axis etc.	10	
Unit: 4	**Windowing & clipping** 4.1 Viewing transformation, Normalization transformation 4.2 Line clipping: Cohen-Sutherland Line clipping algorithm, midpoint subdivision algorithm 4.4 Polygon clipping: Sutherland – Hodgeman Polygon clipping algorithm.	06	
Unit: 5	**Curves** 5.1 Curve generation: Lagrange Interpolation curves, 5.2 B-Spline, Bezier curves.	07	
Unit: 6	**Projection** 6.1 Different Parallel projection 6.2 Perspective Projection.	03	
Total		**45**	

Text Books:

Name of Authors	Title of the Book	Edition	Name of the Publisher
Hearn &Beakar	Computer Graphics through C	5^{th}	Pearson
Pakhira	Computer Graphics Multimedia & Animation	2^{nd}	PHI
Xiang &Plastock	Computer Graphics		McGraw Hill
Maurya	Computer Graphics		Willy

Reference Books:

Name of Authors	Title of the Book	Edition	Name of the Publisher
Kanetkar	Graphics under C		BPB
Udit Agarwal	Computer Graphics		Katson Books

Suggested list of Laboratory Experiments:

Practical

Practical:

Skills to be developed:

Intellectual skills:

- Use of programming language constructs in program implementation.
- To be able to apply different logics to solve given problem.
- To be able to write program using different implementations for the same problem
- Study different types of errors as syntax semantic, fatal, linker & logical
- Debugging of programs
- Understanding different steps to develop program such as

  ☐Problem definition

  ☐Analysis

  ☐Design of logic

  ☐Coding

  ☐Testing

  ☐Maintenance (Modifications, error corrections, making changes etc.)

Motor skills:

Proper handling of Computer System

List of Practical:

1) Implement DDA algorithm for line drawing
2) Implement Bresennham's algorithm for line drawing.
3) Implement Mid-point circle drawing algo.
4) Implement Bresennham's algorithm of circle drawing.
5) Implement Flood fill algorithm for Polygon filling.
6) Implement scan-line algorithm for polygon filling.
7) Write Program for 2-D transformations -> scaling, Rotation,
8) Write Program for 2 D transformations shearing and Translation program
9) Write and implement program for rotation about an arbitrary point.
10) Implement Cohen- Sutherland algorithm for line clipping.
11) Implement mid point subdivision algorithm for line clipping.
12) Implement Sutherland-Hodgeman algorithm for polygon clipping.
13) Write a program to draw a curve using Bezier's algorithm.
14) Write a program to draw curve using B spline.

** Any Graphics program can be done in laboratory (like animation, fractals etc.)

Suggested list of Assignments / Tutorial:

Sl. No.	Topic on which tutorial is to be conducted (To be given as per Lab experiment list)
1.	

Note:

WEB Page Development (Professional Practice - II)

Name of the Course: WEB Page Development (Professional Practice - II)	
Course Code:	**Semester:** FOURTH
Duration: Six months	**Maximum Marks: 50 (Practical)**
Teaching Scheme	**Examination Scheme**
Theory: nil	Mid Semester Exam: Nil
Tutorial: nil	Assignment & Quiz: Nil
Practical: 2 hrs./week	End Semester Exam: 50 Marks (Internal)
Credit: 2	

Aim:	
Sl. No.	
1.	To exploring your business worldwide and makes strong impact image using active online presences with web site. And well-designed and aesthetically appealing website can give you a strong advantage over other online competitors.
2.	To make an interesting to see graphic designers on one end, and web programmers on the other, arguing their respective positions active web page designing is today's need.
3.	To get strong instantaneous recognition of relevance which leads to clarity, and understanding at a glance a well crafted brand strategy which provides context and perspective, and a detailed website plan that spells out specific objectives, target audiences, paths to conversion and other critical elements of your site.

Objective:	
Sl. No.	Students will able to:
1.	Design simple Web pages - using HTML
2.	Organize information using Tables, collect information from users using forms & present information using Frames.
3.	Use style sheets to gain full control of formatting within Web page.
4.	Include ASP within Web pages.
5.	Embed multimedia to Web pages.
6.	Integrate all above to develop Web sites.

Pre-Requisite:	
Sl. No.	The student will be able to:
1.	Interaction with DOS / Windows Operating System.
2.	Ability to develop logic / flow of simple problem.
3.	Web page design tags of Markup language.

Contents	
Sl. No.	Skills to be developed
1.	**Intellectual skills:** ➢ Develop web designing skills. ➢ Apply different logics to solve given problem. ➢ Write program using different interfaces. ➢ Understand client server architecture model and uses. ➢ Embedded programming tricks. ➢ Understanding different steps and stages to develop complex architecture of the WebPages
2.	**Motor skills:** ➢ Proper handling of Computer System**.**

DETAIL COURSE CONTENT (Sessional / Practical)		
Unit	**Contents**	**Remarks**
1	**INTERNET BASICS:** • Familiarity with internet browser (Internet Explorer, Netscape Navigator etc.) • Working with browser window tool bar , menu bar • Browsing a given web site address, searching a particular topic through search engines. • Familiarity with E-Mail, sending viewing printing e-mail message. • Use of mailbox (inbox, outbox) in outlook express. Use of attachment facility available in e-mailing.	
2	**WEB SERVER:** • Familiarity with web server – IIS, PWS etc. – Configuring web server – Creating virtual directory.	
3	**INTERNET SERVICES** • Concept and familiarity of various internet services (www, http, ftp, chat etc).	
4	**HTML/XML** • Creating simple HTML & XML file, place it in web server and access it from client Browser. • Creating a HTML form incorporating GUI components (Command button, text box, radio button, check box, combo box etc).	
5	**ACTIVE SERVER PAGES / ASP.NET** • Introduction to Active Server Pages. • Elements of ASP (Scripts, Objects, Components). • Making your first Active Server Page.	
6	**INTRODUCING VB SCRIPT**: • Variables, Mathematical operators, functions — Logical operators, Loop, Conditional statements — String Function, Date and Time Function. • Subroutine — Formatting Display, Adding Components to scripts — Handling Event driven programming.	
7	**WORKING WITH ASP & ASP.NET**: • Using HTTP — Writing simple ASP files — Controlling Execution of server side scripts. • Problems on HTML forms to get user information and retrieving HTML form contents • Working with query string.	
8	**ASP SESSION:** • Introduction to session. • Familiarity and working with session objects (simple problems). • Using session events. • Familiarity and working with cookies.	
9	**ASP APPLICATION**: • Introduction to ASP Application features of ASP Application • Creating a Simple ASP Application, Setting the properties of ASP Application — Using Application objects and Application events.	
Unit	**Contents**	**Remarks**
10	**ASP COMPONENTS:**	

	• Using Components in ASP (Simple problems) — Creating Components with page scope, session scope, Application scope. • Working with browser capability component, file assess components , counter components etc.(Simple problems)	
11	**DATABASE MANAGEMENT THROUGH ASP:** • Brief overview of ActiveX Data Objects. • Using ADODB to access a database from ASP (Simple Problem) — Opening, closing database connection • Executing SQL statements.	

A sample List of Practical / Sessional works to be done (Leading '*' denotes the harder problems)

S. No.	Specific problem(s) related with Practical / Sessional work	Skill area
01	1.1. Create a static web pages using simple related tags like body with background colour, picture etc., align, font, br etc. 1.2. Embed an image within the page using Src, height, width, border, align, alt etc. 1.3. Implement hyperlinking between two html pages. 1.4. Implement a table with size 4 X 4 on a page and insert some textual as well as numeric data into the cells. Use proper tags for alignment. 1.5. Create a Web page for the following: WELCOME TO XYZ COLLEGE OF ENGINEERING (scroll Horizontally) STUDENT DETAILS (Blink)	HTML

S. No.	S. Name	BRANCH /SEM	Address	Marks		
				M1	M2	M3

01	1.6. Implement frame to display multiple pages on screen.	HTML
02	2.1. *Design Login form with validation. 2.2. *Design Registration form with validation of email address, date of birth, blank field, telephones and mobile numbers etc. 2.3. Design registration form of college using text box, text area, radio list, check list, button etc.	HTML Forms with Scripts.
03	3.1. Apply simple application VBscripts using variables, arrays etc. 3.2. Implement a VBprocedure Sub/ Function to display the area of a circle. Radius of the circle should be passed as a parameter to the procedure. 3.3.* Implement Loop(s) and conditional statement (s) to display all prime numbers between n1 to n2 integral values.	VB Scripting Language
04	4.1. Create an application using ASP to customize a Web Page. 4.2. *Create a login page with user_id and password field that will check whether a user is valid or not. If the user is valid then Loginsuccess page will be displayed otherwise Loginunsuccess page will be generated. 4.3. *Create a short project regarding the maintenance of login page. It should detect an existing user, displays invalid user_id and/or password. Create a new user, update information of an existing user etc.	ASP and its interface with Database

Text Books:

Name of Authors	Title of the Book	Edition	Name of the Publisher
N.P. Gopalan, J. Akilandeswari	Web Technology, A developer's Perspective		PHI
Koggent Learning Solutions	Web Technology (including HTML,CSS,XML,ASP,JAVA) Black Book		Dreamtech
Uttam K Roy	Web Technologies		OXFORD
Ivan Bayross	Practical ASP		BPB

**** During end semester examination all Lecturers should be present.**