SCHEME

Sem	Course code	Course	No. of hrs/week	No. of credits	Hrs/ Sem.	CE	ESE
1	PC1171	Computer Fundamentals and Organization	3	3	54	20%	80%
2	PC1221	Introduction to Programming	3	2	54	“	“
2	PC1271	Programming in C Lab	S1- 2 S2 - 2	3	S1-36 S2-36	“	“
3	PC1371	Microprocessors and Programming	3	3	54	“	“
3	PC1372	Data Structures	3	3	54	“	“
4	PC1471	Software Engineering	3	3	54	“	“
4	PC1472	Object Oriented Programming using C++	2	2	36	“	“
4	PC1473	Programming C++ Lab	S3-2 S4-2	4	S3-36 S4-36	“	“
5	PC1571	Database Management System	3	3	54	“	“
5	PC1581 PC1581.A PC1581.B PC1581.C	Open Course A. Digital Marketing B. Internet and WWW C. Cyber Security	3	2	54	“	“
5	PC1572	PHP and MySQL Lab	7	4	136	“	“
6	PC1671	Computer Networks and Security	4	4	72	“	“
6	PC1672	Operating System	3	3	54	“	“
6	PC1673	Major Project	5	4	90	“	“

Division of Marks (Lab Examination)

(Computer Science)

1. First program should be sufficiently simple – **25 marks**
   (Logic – 10 marks, Successful compilation – 10 marks, Result – 5 marks)
2. Second program should be based on advanced concepts - **30 marks**
   (Logic – 15 marks, Successful compilation – 10 marks, result – 5 marks)
3. Viva Voce - **15 marks**
4. Lab Record - **10 marks**

Total Marks - 80 marks

SPOKEN TUTORIAL SUBJECTS

S.No	Course code	Course	Spoken tutorial courses
1	PC1171	Computer Fundamentals and Organization	Introduction to Computers
2	PC1221	Introduction to Programming	C and CPP
3	PC1372	Data Structures	Advance C
4	PC1472	Object Oriented Programming using C++	Advanced CPP
5	PC1571	Database Management System	PHP and MySQL

PC1171: COMPUTER FUNDAMENTALS AND ORGANIZATION

1. AIM

- To create the overall generic awareness about the field of Information Technology and to impart knowledge in the functional organization of physical components and architecture of a computer.

2. OBJECTIVES

On the completion of this course, the student will be able:

- To get the basic concepts of Computers.
- To get the functional knowledge about PC hardware, operations and concepts.
- To understand the functional units of a standard PC and it’s working.
- To understand the memory organization in a computer.

3. SYLLABUS

Module I: Characteristics of Computer; Von Neumann model; Inside a Computer: SMPS, Motherboard, BIOS, CMOS, Ports and Interfaces, Expansion Cards, Ribbon Cables, ASCII; Types of Input Devices, Types of Output Devices.

Module II: Memory Representation, Hierarchy, Memory Units: RAM (SRAM, DRAM); ROM; Secondary Storage Devices: Magnetic Tape, Magnetic Disk, Types of Magnetic Disks, Optical Disk, Types of Optical Disks; USB: Pen drive, External Hard Disk; Memory Stick; CPU Registers, Cache Memory, Operations in Cache memory, hit ratio; Virtual Memory.

Module III: Instruction Format; Instruction Cycle: Fetch Cycle, Execution Cycle; Instruction Set: CISC Architecture, RISC Architecture, Comparison; Memory Chips; Pipelining and Parallel Processing; Microprogrammed Control and Hardwired Control.

Module IV: Input/Output Organization: Asynchronous Data Transfer, Programmed I/O (concepts only); Interrupts: Types of interrupts, processing interrupts, interrupt hardware and priority, DMA: DMA Controller, DMA Transfer Modes; I/O Processor.

4. REFERENCES

4.1 Core

- Introduction to Information Technology, 2nd Edition, ITL Education Solutions Limited, Pearson.
- John D. Carpinelli, Computer systems Organization & Architecture, Pearson Education.

4.2 Additional

- E.Balaguruswamy, Fundamentals of Computers, McGraw hill, 2014
- Carl Hamacher, Vranesic, Zaky, Computer Organization 4th Edition, McGraw-Hill

4.3 Assignments and Activities: *Applications of Computers in various fields*; *Pioneers in IT*; *IT Policy*, *IT and Development*; *IT in India* (*major initiatives*, *key institutions*, *statistics*), *IT in Kerala* (*major initiatives*, *key institutions*, *statistics*); *Careers in IT*; *Computer faults*: *hardware & software*; *types of faults*; *diagnostic programs and tools*; *printer problems*; *monitor problems*, *problem diagnosis*, *organization of a modern PC*.

NB:- Activities and assignments are not meant for End_Semester_Examination

PC1221: INTRODUCTION TO PROGRAMMING

1. AIM

- To Expose students to algorithmic thinking and problem solving and impart moderate skills in programming in an industry-standard programming language.

2. OBJECTIVES

- To expose students to algorithmic thinking and algorithmic representations.
- To introduce students to basic data types and control structures in C.
- To introduce students to structured programming concepts.
- To introduce students to standard library functions in C language.

3. SYLLABUS

Module I: Introduction to programming: Algorithm & Flow charts: Definitions, Symbols used to draw flowcharts, Program Writing – Structure of the Program, Source code, Object code, Executable file, Variables and Constants, Rules for naming the Variables/Identifiers; Basic data types of C, int, char, float, double; storage capacity – range of all the data types;

Module II: Basic Elements: Operators and Expressions: Expression Evaluation (Precedence of Operators); simple I/O statements, Control structures, if, if else, switch-case, for, while, do-while, break, continue. Arrays: Defining simple arrays, Multi-dimensional arrays, declaration, initialization and processing.

Module III: Functions & Pointers: concept of modular programming, Library, User defined functions, declaration, definition & scope, recursion, Pointers: The & and * Operators, pointer declaration, assignment and arithmetic, visualizing pointers, call by value; call by reference, dynamic memory allocation. Storage classes.

Module IV: Advanced features: Array & pointer relationship, pointer to arrays, array of pointers. Strings: String handling functions; Structures and unions; File handling: text and binary files, file operations, Functions for file handling, Modes of files

4. REFERENCES

4.1 Core

- Ashok N. Kamthene, *Programming in C*, Pearson Education, Second edition

4.2 Additional

- E. Balaguruswamy, *Programming in ANSI C*, McGrawhill, Sixth Edition

4.3 Assignments and Activities: *Pre-processor directives: #include, #define, macros with arguments, the operators# and ##, conditional compilations, multiple file programming; creating header files, program verification, algorithm efficiency analysis; int86 functions and graphic functions.*

NB:- Activities and assignments are not meant for End_Semester_Examination

PC1271: PROGRAMMING IN C LAB

1. SYLLABUS

1. Familiarization of important DOS/Windows/Linux features
2. Practice on basic features of word processor, spread sheet and presentation software.

Part A

The C laboratory work will consist of 25-30 Experiments

1-15.Testing out and interpreting a variety of simple programs to demonstrate the syntax and use of the following features of the language: basic data types, operators and control structures.

Part B

16. 1-D Arrays: A variety of programs to declare, initialise, read, print and process 1-D arrays of various basic data types. Processing to include, selection, sum, counting, selective sum, selective counting, reversing etc.
17. Pointers: A large number of trivial programs involving all possible data types to familiarize the syntax of pointers in a variety of situations and to draw memory diagrams based on the observations.
18. Structures: A variety of programs to declare, initialise, read, print and process structures made up of a variety of data types and structures.
19. 2-D Arrays: A variety of programs to declare, initialise, read, print and process 2-D arrays of various basic data types. Processing to include, selection, sum, counting, selective sum, selective counting, reversing etc.
20. Array of Structures and Structure of Arrays: Programs to demonstrate declaration and processing of structure of arrays and array of structures.
21. Pointers to Arrays: A number of programs to demonstrate handling of 1-D and 2-D arrays using pointers and to draw memory diagrams based on the observations.
22. Pointers to Structures: A number of programs to demonstrate use of pointers to structures and to draw memory diagrams based on the observations.
23. Functions –I: Simple Examples of declaring and using functions of the following categories (i)no argument, no return, (ii) argument, no return, (iii) no argument, return, (iv) argument, return, all pass by value
24. Functions –II: Declaring and using functions with pass by reference, Passing and Returning structures, Recursive functions.
25. Files: Simple Example involving use of multiple files: declaring, opening, closing, reading from and writing to text files.
26. Files: Example involving use of multiple files: declaring, opening, closing, reading from and writing to binary files.
27. Library functions: A variety of Examples demonstrating (i) string processing functions (ii) a variety of selected library functions
28. Debugging programs involving syntactic and/or logical errors

29-30: Developing programming solutions to problems including program design, algorithm development and data structure selection.

PC1371: MICROPROCESSORS AND PROGRAMMING

1. AIM

- To introduce 80x86 assembly language and thereby familiarize the student with architecture of microprocessors

2. OBJECTIVES

By the end of the course, students should be able to:

- Appreciate architectural features of x86 family of processors
- Read and write moderately complex assembly programs for 8086 processor
- Use the tools debug, TASM/MASM, Unix/Linux Code view
- Use assembly routines in C/C++

3. SYLLABUS

Module 1: Microprocessor Literature, Evolution of microprocessor, Basic functional blocks of a microprocessor, Microprocessor based systems, Concept of multiplexing; **Intel 8086 Pins, Signals and architecture:** Introduction, pins and signals, architecture, instruction and data flow, even and odd memory banks, bus cycles and timing diagram

Module 2: Instruction set of 8086: Instruction format, addressing modes, execution time, affecting flags, Data transfer, arithmetic, logical, string manipulation, control transfer and processor control instructions; **Interrupts:** Its need, classification, sources, interrupts of 8086, Implementing interrupt scheme, INTR and its expansion, Programmable Interrupt Controller,

Module 3: Assembly language programming: Program development tools, variables and constants used in assemblers, assembler directives, Procedure and macros, Interrupts of personal computers, Hand coding of assembly language programs, examples

Module 4: 80X86 family of processors: Introduction, 80186, 80286, 80386, 80486, Pentium, Advanced Pentium processors

4. REFERENCES

4.1 Core

- A Nagoor Kani ,*8086 Microprocessor and its applications*, McGrawhill, Second edition

4.2 Additional

- N. Madhivanan, Microprocessors, PC Hardware and Interfacing, PHI Edition
- RS Goankar, *Microprocessor Architecture*, *Programming and Applications with 8086*, Wiley Eastern Edition.

4.3 Assignments and Activities: *Miscellaneous Topics*: *Features of core2*, *dual core and i series Processors*, *RISC*, *CISC*, *Motherboard of IBM PC*, *Drives*, *Peripherals*, *I/O buses*, *Parallel*, *Serial and USB ports*.

NB:- Activities and assignments are not meant for End_Semester_Examination

PC1372: DATA STRUCTURES

1. AIM

- To introduce students to various data structures and their features and applicability.

2. OBJECTIVES

By the end of the course, students should:

- Be able to write well-structured programs in C
- Be familiar with data structures like array, structures, lists, stacks, queues, trees and graphs
- Able to implement the above data structures in C/C++
- Able to appreciate various searching and sorting strategies
- Able to select appropriate data structures for solving a given problem

3. SYLLABUS

Module–I: Sequential searching, binary searching, Hashing – linear hashing, hash functions, hash table searching, Sorting: bubble sort, selection sort, Stacks and Queues: FIFO and LIFO data structures – stacks using (i) pointers and (ii) arrays. Queues using (i) pointers and (ii) arrays, Operations on stack and queues; applications polish notation.

Module- II: Linked Lists: Concept of static versus dynamic data structures, implementation of linked lists using pointers, operations on linked lists: insertion, deletion and traversing. Doubly linked lists and circular linked lists, applications of linked lists.

Module-III: Trees: Concept of linear versus non-linear data structures, various types of trees – binary, binary search trees. Creating a binary search tree, traversing a binary tree (in-order, pre-order and post-order), operations on a tree –insertion, deletion and processing, expression trees, implementation using pointers, applications.

Module–IV: Graphs, graph traversal- depth-first and breadth-first traversal of graphs, applications.

4. REFERENCES

4.1 Core

- A.K.Sharma, *Data Structures Using C*, Pearson, Second edition, 2011

4.2 Additional

- Nair A.S., Makhalekshmi, *Data Structures in C*, PHI, Third edition 2011.

4.3 Assignments and Activities: *Multi-way search trees*, *B-trees*, *Huffman trees*, *case studies*.

NB:- Activities and assignments are not meant for End_Semester_Examination

PC1471: SOFTWARE ENGINEERING

1. AIM

- To introduce the basic concepts of software engineering

2. OBJECTIVES

At the end of the course, the students should be able to

- Understand the importance of basic processes in software Development life cycle.

- Understand the various activities incorporate with different models and know their significance.
- Familiarize the requirements in engineering and systematic approach in classical software design and development techniques.
- Familiarize with various software testing techniques and tools.

3. SYLLABUS

Module I: Introduction: Evolution; **Software life cycle models**: A few basic concepts, Waterfall model and its extension, Agile development models, Spiral model, Comparison of different life cycle models

Module II: Software Project Management, Project Planning, Metrics for project size estimations, Project Estimation Techniques, Basic COCOMO model, Risk Management, Software **Requirements Analysis and Specification**: Requirements gathering and analysis, Software Requirements Specification

Module III: Software Design: overview of the design process, How to characterise a good software design, Cohesion and Coupling, Approaches to software design, **Function oriented design**: Overview of SA/SD Methodology, Structured analysis, Developing the DFD model of a system, Structured Design, **User Interface design**: Characteristics of a good user interface, Basic concepts, Types of user interfaces

Module IV: Coding and Testing: Coding, Code review, Software documentation, Testing, Unit testing, Black box testing, white box testing: Basic concepts, Debugging Integration testing, system testing, **Software Reliability and quality management**: Software reliability, Software quality, **Software maintenance**: Characteristics of software maintenance, Software reverse engineering, **Emerging Trends:** Client Server Software, Client Server architectures, CORBA, Service Oriented Architectures (SOA), Software as a Service.

4. REFERENCES

4.1 Core

- Rajib Mall, *Fundamentals of Software Engineering*, Fourth Edition, PHI

4.2 Additional

- Pankaj Jalote, *An Integrated Approach to Software Engineering*, Narosa
- Waman S. Jawadekar, *Software Engineering*, McGraw hill, 2013
- Software Engineering (Seventh edition), Ian Sommerville – Addison Wesley.
- Software Engineering: A practitioners approach (Sixth Edition), Roger S Pressman-Mc Graw Hill.
- *Journals and Magazines*: (i) Software Development, CMP Media. (ii) Software Quality Professional, ASQ.

4.3 Activities and Assignments: *Preparing various documents*, *case studies*, *preparing test plans*, *UML diagrams*, *Metrics for various development phases*, *Agile Programming Methodologies*, *extreme Programming*, *Formal Methods*, *CASE Tools*.

NB:- Activities and assignments are not meant for End_Semester_Examination

PC1472: OBJECT ORIENTED PROGRAMMING USING C++

1. AIM

- To introduce the student to the basic concepts of object orientation and impart skills in an Industry standard object oriented language

2. OBJECTIVES

On the completion of this course, the student will be able to

- Understand the concepts of classes and object
- Define classes for a given situation and instantiate objects for specific problem solving
- Reuse available classes after modifications if possible
- Possess skill in object oriented thought process

3. SYLLABUS

Module I: **Concepts of OOP**: Introduction OOP, Procedural Vs. Object Oriented Programming, Principles of OOP, Benefits and applications of OOP, Object Oriented Concepts, reusability, maintenance, security, comfort in programming. **C++ Basics**: Overview, Program structure, Basic data types and declarations, operators, control structures. **C++ Functions**: Simple functions, Call and Return by reference, Inline functions, Macro Vs. Inline functions, default arguments.

Module II: **Classes and objects** in C++, access modifiers, static data members and member functions, friend functions and friend class, Constructors and Destructors – characteristics, applications, overloading, copy constructors, Overloading of functions, Operators Overloading- Unary and Binary, Overloading rules, Type conversion.

Module III: **Inheritance**- parent and child classes, access specifiers - private, public and protected inheritance, types – single, multiple, hierarchical, multi-level inheritance, advantages and disadvantages of inheritance. Virtual base classes. Dynamic memory allocation using new and delete operators, dynamic objects.

Module IV: **Binding & Polymorphism**: Early binding, Late Binding, Pointers to derived class objects, virtual functions, Pure virtual functions, abstract classes, object slicing, **I/O and File management**: Concept of streams, cin and cout objects, C++ stream classes, Unformatted and formatted I/O, manipulators, File stream, C++File stream classes, File management functions, File modes, Binary and random files, Exception handling in C++: try, throw and catch.

4. REFERENCES

4.1 Core

- Ashok N. Kamthane, *Object oriented Programming with ANSI & Turbo C++*, Pearson

4.2 Additional

- H M Deitel and P J Deitel, *C++: how to program*, Pearson Education
- Robert Lafore, *Object Oriented Programming in Turbo C++*, Galgotia Publications

5. ASSIGNMENTS AND ACTIVITIES: Evolution of OOP – history of C and C++, Review of features of C++ common with C and also minor variations; study of File stream classes in C++. Templates class and function templates, Templates versus macros, String objects in C++, Standard Template Library in C++.

NB:- Activities and assignments are not meant for End_Semester_Examination

PC1473: PROGRAMMING IN C++ LAB

The laboratory work will consist of 15-20 experiments, only by using class concept

Part A

1. Testing out and interpreting a variety of simple programs to demonstrate the syntax and use of the following features of the language: basic data types, operators and control structures.
2. Solving a problem using (i) structures and (ii) classes and comparison between the two (the problem logic and details should be kept minimal and simple to enable focus on the contrast between the two methods, for example declaring result of a set of students defining the name and total marks in the program itself).
3. Class definitions and usage involving variety of constructors and destructors

Part B

4. Programs involving various kinds of inheritances,
5. Programs involving operator overloading and type conversions
6. Programs involving virtual base classes, friend functions
7. Program to demonstrate early and late binding
8. Program to allocate memory dynamically
9. Program involving class and function templates
10. Programs to demonstrate(i) string processing (ii) file streams (iii) a variety of selected library functions
11. exception handling
12. Handling of 2-D arrays using pointers
13. Debugging programs involving syntactic and/or logical errors

PC1571: DATABASE MANAGEMENT SYSTEM

1. AIM

- To introduce basic concepts of data bases, and related techniques and tools

2. OBJECTIVES

- Be aware of basic concepts of data bases and data base management systems
- Be aware of concepts of relational data bases.
- Know to normalize relational data bases
- Skilled in using relational algebra and relational calculus
- Develop skills to write database queries

3. SYLLABUS

Module I: Introduction: evolution of data base systems, overview of database management systems, Relational data model, mathematical definition, candidate, primary and foreign keys, set operations on relations, insertion, deletion and update operations, attribute domains.

Module II: The E-R Model, Entities and attributes, 1-1 and many-1, many-many relationships. Security – Physical and Logical, Design and maintenance issues, integrity.

Module II: Relational algebra and relational calculus, Introduction to SQL, Table creation, selection, projection and join using SQL

Module IV: Functional Dependencies – Inference axioms, Normalization, 1NF, 2NF, 3NF and Boyce - Codd Normal forms, Lossless and lossy decompositions.

4. REFERENCES

4.1 Core

- Ramon A. Mata-toledo and Pauline K. Cushman, *Fundamentals of Relational Data Bases*, Schaum Outlines, Tata McGraw Hill

4.2 Additional

- Atul Kahate, *Introduction to Data Base Management Systems*, Pearson Education

4.3 Assignments and activities: *Study of features of MS Access, Open Office Base, Oracle, mySQL, emerging areas.*

NB:- Activities and assignments are not meant for End_Semester_Examination

PC1581: OPEN COURSE

PC1581.A DIGITAL MARKETING

1. AIM

- To introduce the student to the basic concepts of digital marketing functions

2. OBJECTIVES

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

- To familiarize students with Digital marketing function in organizations.
- To understand different modes of payments, beware of security and legal issues in digital marketing

3. SYLLABUS

Module I: Introduction: Nature, Scope and Importance of Digital Marketing; Evolution of Digital Marketing; Core Concepts-Inbound Marketing, Content Marketing, Email Marketing, Influential Marketing; Holistic Digital Marketing Concept, 10Ps of digital marketing; Digital Marketing Environment: Macro and Micro Environment.

Module II: E-banking: approaches, devices, services, benefits, drawbacks, Electronic payment systems-credit cards, debit cards, smart cards, credit accounts, cyber security, encryption, secret key cryptography, public key cryptography, digital signatures, firewalls

Module III: Digital Marketing: Search Engine Optimization (SEO), Social Media, Content Marketing; Email Marketing, Mobile Marketing. Challenges for Digital Marketing: Increased Security Risk, Cluttered Market, Less Focus on Keywords, More Ad Blockers, Increased Ad Costs**.**

Module IV: Digital Marketing: Pay per Click-Search Engine Advertising, Advantages, Factors, Conversion Rate Optimization (CRO); Digital Marketing- Web Analytic. Social Media Marketing: Face book, Pinterest, Twitter, LinkedIn, YouTube, Google Adwords, Google Analytics; Issues and Future enhancement of Digital Marketing.

4. REFERENCES:

4.1 Core

- Ian Dodson-*The art of Digital Marketing*, Wiley

4.2 Additional

- Puneet Singh Bhatia- *Fundamentals of Digital Marketing*, Pearson Education

4.3 Assignments and activities: ***Collection of current marketing tools, case studies, new trends.***

NB:- Activities and assignments are not meant for End_Semester_Examination

PC1581.B INTERNET AND WWW

1. AIM:

- To introduce to Internet and World Wide Web.

2. OBJECTIVES:

- To understand the basic concepts of Networks.
- To learn the working of Internet.
- Exposure to Network Protocols and WWW

3. SYLLABUS

Module I: Introduction to Network-Types of Network-Network Topologies, Intranet, Extranet-Advantages, Internet-History, Modes of Connecting to Internet-Dialup Access-Direct to dedicated connections, Internet Service Providers, Domain Name Service, Internet Addresses-Addressing Scheme-IPV6,Modems,Communication Software, Internet tool-File Transfer Protocols, Search Tools, Telnet.

Module II: Introduction to WWW-WWW and HTTP, Webpage, Introduction to Web Browser-Book Marks-Comparison, Directories, Search Engines-Search Fundamentals-Search Terminology-Search Strategies – Search Generalization-Search Specialization-Working.

Module III: Uniform Resource Locator (URL) Introduction to TCP/IP-TCP/IP Model, Email-Working with Email-Sending Mail-Reading Mail-Replying to Mail-Deleting Mail-Advantages and Disadvantages of Email, Basics of Chat Rooms, SMTP.

Module IV: Introduction to Web Server-Personal Web Server (PWS)-Internet Information Server (IIS)-Apache Webserver-Benefits of Web Server, Introduction to Security-Internet Security-Identifying Network Stations, Network Protocols-Internet Security Threats.

4. REFERENCES

4.1 Core

- Dr.Surender Jangra, “Basics of Internet and Web”, Vayu Education of India. New Delhi 110002

4.2 Additional

- Raymond Greenlaw, Ellen Hepp, “Fundamentals of Internet and the World Wide Web, McGraw-Hill.

NB:- Activities and assignments are not meant for End_Semester_Examination

PC 1581.C CYBER SECURITY

1. AIM

- To introduce the issues and methods of information security and its guidelines.

2. OBJECTIVES:

On completion of this course student shall:

- Understand high-level overview of information security principles.
- Understand different roles and responsibilities of security professionals
- Understand cryptography and information system risk management.
- Be aware of multiple security control families as well as benefits of each control family

3. SYLLABUS

Module I: Introduction to Information systems: Modeling business process, components, categories, Individuals in information system, Developing information systems; **Information Systems:** threats, Information assurance, cyber security and security risk analysis; **Application security:** Data Security considerations, security technology, intrusion detection access control

Module II: Security threats: Introduction to security threats, Network and services attack, security threats to e-commerce

Module III: Security Policies: Introduction, Why we need security policies, Security policy development, Email security policies, Policy review process, corporate policy, sample template of cyber security policy

Module IV: Information security standards: Why ISO, IT ACT 2000, copyright, Patent, Intellectual property right, Cyber laws in India, Software Licensing, Semi conductor law and patent law,

4. REFERENCES

4.1 Core

○ Fundamentals of Cyber security, Mayank Bhushan, BPB publication, First Edition 2017

NB:- Activities and assignments are not meant for End_Semester_Examination

PC1572: PHP and MySQL LAB

The laboratory work will consist of 15 -20 Experiments

Part A (MYSQL)

- Database creation, table creation, insertion, updation, deletion and select.
- Programs to connect PHP and MYSQL
- Setup WAMP/XAMPP Server or Setup Apache, MySQL and PHP separately in your PHP Lab.

Part B (PHP)

- Write a PHP program to generate a random number between 1 and 100.

- Modify above program to accept range of the random number from HTML interface.
- Programs involving various control structures like if, else, elseif/else if', Alternative Syntax for 'if, else, elseif/else if'
- Programs involving various control structures like while, do-while, for, foreach, switch, break, continue. Try alternative syntax for while, do-while, for, foreach, switch.
- Programs involving the declaration, return, require, include, require- once, include_once and goto.
- Programs to demonstrate PHP Array functions, PHP Array Sorting, PHP Key Sorting, PHP Value Sorting, PHP Multi Array Sorting, PHP Array Random Sorting,
- Programs to demonstrate PHP Array functions. PHP Array Reverse Sorting, Array to String Conversion, Implode() function, String to Array, Array Count, Remove Duplicate Values
- Programs to demonstrate PHP Array functions. array Search, Array Replace, Array Replace Recursive, Array Sub String Search
- Demonstrate the use of regular expression to compare two strings.
- Extract Domain name from URL
- Find the number of rows from a MYSQL database for your query.
- Generate a Guestbook which will allow your website visitor to enter some simple data about your website.
- Develop a PHP program for Email Registration.
- Develop a project for making Application form and performing Degree Admission On-line.

[Sample Questions]

- Write a PHP script to find the factorial of a given number.
- Write a PHP script to find the sum of digits of a given number.
- Write a PHP script to find whether the given number is a prime or not.
- Write a PHP script to demonstrate the use of break, continue statements using nested loops.
- Write a PHP script to display the Fibonacci sequence with HTML page.
- Write a PHP script to create a chess board.
- Write a PHP script using built-in string function like strstr(), strops(), substr_count(), etc
- Write a PHP script to transform a string to uppercase, lowercase letters, make a string's first character uppercase.
- Write a PHP script that inserts a new item in an array in any position.
- Write a PHP function to check whether all array values are strings or not.
- Write a PHP script to count number of elements in an array and display a range of array elements.
- Write a PHP script to sort a multi-dimensional array set by a specific key.
- Write a PHP script using a function to display the entered string in reverse.
- Write a PHP script using function for sorting words in a block of text by length.
- Write a PHP script for creating the Fibonacci sequence with recursive function.
- Write a PHP script using pass by value and pass by reference mechanisms in passing arguments to functions.
- Write a PHP script to demonstrate the defining and using object properties.
- Write a PHP script to demonstrate the inheritance.
- Write a PHP script to demonstrate the object overloading with _get(), _set(), and _call().
- Write a PHP script to demonstrate the overloading property accesses with _get() and _set().
- Write a PHP script to demonstrate the object overloading with _get(), _set(), and _call().
- Write a PHP script to demonstrate the overloading property accesses with _get() and _set().
- Write a PHP script to demonstrate the use interfaces.
- Write a PHP script using constructors and destructors.
- Write a PHP application to handling HTML forms with PHP script.
- Write a PHP script to create a file, write data into file and display the file's data.
- Write a PHP script to check and change file permissions, copying, renaming and deleting files.

- Write a PHP application for connecting to MySQL and reading data from database table.
- Write a PHP application for inserting, updating, deleting records in the database table.
- Write a PHP application for student registration form.

PC1671: COMPUTER NETWORKS AND SECURITY

1. AIM

- To introduce students to basic functions and the theoretical underpinning's of modern operating systems

2. OBJECTIVES

To introduce students to:

- Fundamental concepts of systems software and functions of operating systems as a resource manager
- Strategies for constrained resource allocation and process scheduling
- Memory and I/O Management techniques
- Salient features of popular operating systems.

3. SYLLABUS

Module I: Introduction to operating system: Introduction, Operating system structures-Operating System Operations, operating system services, user operating system interface, system programs, system calls, Types of System Calls, operating system structure. **Process Management**: Process concept, Process Scheduling, Operations on processes, Inter-process communication, Threads-Overview, Multithreading model, Thread Libraries, Threading issues; CPU Scheduling: Basic concepts, scheduling criteria, Scheduling algorithms.

Module II: Process synchronization: Background, Critical section problem, Peterson's solution, Semaphore, Classical synchronization problem-bounded buffer problem, reader/writer problem. The Dining Philosophers problem.**Deadlocks:** deadlock characterization, methods for handling deadlock-deadlock prevention, deadlock avoidance, deadlock detection, deadlock recovery.

Module III: Memory Management & Protection: Basic Hardware, Address binding, Logical versus physical address space, Swapping, Contiguous memory allocation- memory mapping and protection, memory allocation, fragmentation. Non-contiguous allocation – paging, segmentation. **Virtual memory**- Demand Paging, page replacement, Allocation of Frames, Thrashing, Allocating Kernel Memory. Protection and Security: Protection -principles of protection, domain of protection, access matrix, access control; Security- threats, user authentication.

Module IV: Storage management: File system Interface - file concept, access methods, directory structure, File Sharing, **File system implementation**- file system structure &implementation, directory implementation, allocation methods, free space management; Mass storage management - disk structure, disk scheduling, RAID;I/O Systems – I/O hardware, Application I/O interface, kernel I/O subsystem.

4. REFERENCES

4.1 Core

- Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, *Operating System Concepts*7th edition

4.2 Additional

- Achyut S Godbole, *Operating systems*, McGRawhill, Third Edition

4.3 Assignments and Activities: *case study of popular Operating Systems like Android, Windows, Sun Solaris, IOS etc*

NB:- Activities and assignments are not meant for End_Semester_Examination

PC1672: OPERATING SYSTEM[redacted]

1. AIM

- To introduce computer networks as well as methods of information security.

2. OBJECTIVES

On completion of this course student shall be able to understand:

- The basic transmission technologies and characteristics
- The use of layer architecture for networking systems
- The main design issues of transport protocols and the mechanism to control traffic flow and congestion.
- The concept of Information security policies

3. SYLLABUS

Module I: Introduction to networks – Data Communication – Data flow simplex, Half duplex, Full duplex-Type of Connection – broadcast, Point–to-Point, multi-drop. Bandwidth- bit rate, baud rate. Transmission media –Copper wires, fibre optics, Radio transmission, microwave, Satellite. Switching - circuit, packet, message.

Module II: Network software - standards- Layering, packets, Layered PDUs, ISO-OSI model, TCP/IP model – Comparison. Framing- bit oriented, byte oriented, Error correction – detection – parity, hamming code, CRC. Flow control – stop and wait, sliding window, Error control - Stop &wait ARQ, Go Back N ARQ, Piggybacking, pipelining, Ethernet, Multiple Access Protocols - pure- slotted ALOHA, CSMA, CSMA/CD. LAN Standards, Token bus, Token ring, Interfacing devices – bridge, hub, switch, router, gateway.

Module III: Internetworking- datagrams, fragmentation – Routing-Flooding, Distance vector routing, Link state routing. Concepts of congestion control-leaky bucket algorithm. Process to Process delivery -TCP, UDP, Application Layer -DNS, Remote login, file transfer protocol (FTP).

Module IV: Information Security: Network security – concepts and policies, cryptography -encryption, ciphers, steganography, symmetric and public key encryption, RSA algorithm, authentication methods, message digest, digital signatures, DSS, E-mail security, MIME, IP Security, Web Security: Secure Socket layer. Malicious Software, viruses & anti-virus software, firewall. Security and Law: - Regulations in India. Indian Copyright Act, Consumer Protection Act. Future Trends – The Law of Convergence.

4. REFERENCES

4.1 Core

- Brijendra Singh, Data Communication and Computer Networks, 3/e, PHI
- Brijendra Singh, *Cryptography & Network Security*, PHI.
- Pachghare, V.K., *Cryptography and Information Security*, PHI

4.2 Additional

- Behrouz A Forouzan, Data Communication and Computer networks, 4th ed,McGraw Hill
- Achyut S Godbole, Data communications and networks, McGrawHill, Second
- Tanenbaum, “computer-networks-a--4th-edition”

4.3 Assignments and Activities: *AES, Blowfish algorithms, Kerberos, Comparison of PGP and/ SMIME. Study of common malicious software, Anti-viruses. Firewall, Trusted systems. Information Technology Act2000/2008.*

NB:- Activities and assignments are not meant for End_Semester_Examination

PC1673: MAJOR PROJECT

1. **AIM**

- To expose student to industry-standard project practices, through a real-life project work under time and deliverable constraints, applying the knowledge acquired through various courses.

2. **OBJECTIVES**

- To provide an opportunity to apply the knowledge gained through various courses in solving a real life problem
- To provide an opportunity to practice different phases of software/system development life cycle
- To introduce the student to a professional environment and/or style typical of a global IT industry
- To provide an opportunity for structured team work and project management
- To provide an opportunity for effective, real-life, technical documentation
- To provide an opportunity to practice time, resource and person management.

3. **PROJECT GUIDELINES**

- Group Size – Maximum 4, most preferably- 3
- No. of records – No. of group members+ 1 (Department copy)
- Certificate should include the names of all members

The minimal phases for the project are: Project feasibility, Investigation of system requirements, Data and Process Modeling, System Design, Program design, Program coding and unit testing, System integration, System implementation and acceptance testing.

3.1 Planning the Project: The Major Project is an involved exercise which has to be planned well in advance. The topic should be chosen in Semester 4 itself and the study of Course CS1342 should as far as possible, be based on the project topic, although in cases with valid reasons, the project guide may waive this condition. Related reading, training and discussions should start from semester 5 itself.

3.2 Selection of project work: Project work could be of 3 types:

a) Developing solution for a real-life problem: In this case, a requirement for developing a computer based solution already Exists and the different stages of system development life cycle is to be implemented successfully. Examples are Accounting Software Package for a particular organization, Computerization of administrative functions of an organization, Web Based Commerce, etc. The scope for creativity and exploration in such projects is limited, but if done meticulously, valuable experience in the industrial context can be gained.

(b) Innovative Product development: These are projects where a clear-cut requirement for developing a computer based solution may not be existing, but a possible utility for the same is conceived by the proposer. An Example is a Malayalam Language Editor with Spell Checker, Computer Music Software for Indian Music, Heat Engines Simulation Software for eLearning, Digital Water Marking Software etc.

(c) Research level project: These are projects which involve research and development and may not be as structured and clear cut as in the above case. Examples are Malayalam Character Recognition, Neural Net Based Speech Recogniser, Biometric Systems, Machine Translation System etc. These projects provide more challenging opportunities to students and can be attempted.

If any student identifies proper support in terms of guidance, technology and references from External organizations and also the supervisors are convinced of the ability of the student(s)to take up the project, it shall be permitted. The methodology and reporting of such projects could be markedly different from type (a) and is left to the proposer/external supervisor of the projects.

3.3 Selection of Team: To meet the stated objectives, it is imperative that Major Project is done through a team effort. Though it would be ideal to select the team members at random (drawing lots) and this should be strongly recommended, due to practical considerations, students may also be given the choice of forming themselves into teams preferably **3** in numbers up to a maximum of **4** members (teams less than 3 members may be permitted in certain cases, for valid reasons). A gender mix should also be strongly suggested. A team leader shall be elected through drawing lots. Teams shall maintain team meeting minutes and ensure that every team member has tasks assigned in writing. Team meeting minutes shall form a part of the Project Report. Even if students are doing projects as groups, each one must independently take up different modules of the work and must submit the reports also independently (though, in such cases, some common materials is permissible).Evaluation will also be done independently.

3.4 Selection of Tools: No restrictions shall be placed on the students in the choice of platforms/tools/languages to be utilized for their project work, though open source is strongly recommended, wherever possible. No value shall be placed on the use of tools in the evaluation of the project.

3.5 Selection of Organization & Guide: No restrictions shall be placed on the students in the choice of organization where project work may be done, in terms of locality, type (public/private) etc. It is the duty of the Head of Institute/Principal of College to ensure that the Aim, Objectives and full project guidelines are communicated to the external organization. The guide should ideally be a post-graduate with minimum 2 years of work experience.

Students may also choose to do project in the college/institute (or partially in the college/institute and partially in an external organization), especially product-based work, but in such cases the supervisors must ensure that (i) industry practices are followed (ii) the students undertake a planned visit to an IT industry with international operations to make up for the loss of experience and (iii) the services of an external guide with industry experience is obtained.

3.6 Project Management: Head of Department /Institute should publish a list of students, projects topics, internal guide and external organization (if any) and teams agreed, before the end of semester 5. Changes in this list may be permitted for valid reasons and shall be considered favourably by Head of Department /Institute any time before commencement of the project. Any request for change after commencement should considered by a committee of 3 teachers and their recommendation shall be accepted by Head of Department/ Institute.

Gantt-chart of proposed activities and a draft statement of project deliverables (which may subsequently be altered if justified) should be prepared before the commencement of the project. The actual completion of each phase should be noted on the chart in the course of the project work. Team meetings should document the progress of the project. Students should submit a fortnightly report of progress which could be indication of percentage of completion marked on the original Gantt-chart, with any notes attached. Students should ideally keep a daily activity log sheet. Changes in the submitted documents are possible, as project development is essentially an evolutionary process. The project guide must ensure that changes are necessary due to the knowledge gained in succeeding phases of the project. The date of Completion of a phase should be brought forward if the changes made are deemed to be errors and not due to additional knowledge gained from a succeeding phase.

3.7 Documentation:

The following are the major guidelines: The final outer dimensions of the report shall be 21 cm X 30 cm. The colour of the flap cover shall be light green. Only hard binding should be done, with title of the Project and the words "< TITLE> BSc(CS) Project Report 2018" displayed on the spine in 20 point, Bold, Times New Roman. It is highly recommended that Latex be used for documentation.

- The text of the report should be set in 12 pt, Times New Roman, 1.5 Spaced.
- Headings should be set as follows: CHAPTER HEADINGS 20 pt, Times New Roman, Bold, All Caps, Centered.

1. SECTION HEADINGS 12 pt, Times New Roman, Bold, All Caps, Left Adjusted.

1. 1 Section Sub-headings 12 pt, Times New Roman, Bold, Left Adjusted.

Titles of Figures, Tables etc are done in 12 point, times New Roman, Italics, Centered.

<PROJECT TITLE>

<STUDENT NAME>

<COLLEGE NAME and EMBLEM>

PROJECT REPORT

Submitted in partial fulfilment of the
Requirements for the award of
BSc (Physics and Computer Applications) degree of
University of Kerala

2018

Some general guidelines on documentation stylistics are:

- Double quotes and single quotes should be used only when essential. Words put in quotes are better highlighted by setting them in italics. Eg: This process is known as "morphing". This process is known as *morphing.*
- Page numbers shall be set at right hand top corner, paragraph indent shall be set as 3.
- Only single space need be left above a section or sub-section heading and no space may be left after them.
- Certificate should be in the format: "Certified that this report titled...................... is a bonafide record of the project work done by Sri/Kum...................... under our supervision and guidance, towards partial fulfillment of the requirements for the award of the Degree of BSc (Computer Science) of the University of Kerala" with dated signatures of Internal Guide, external guide and also Head of Department/Institute.
- If the project is done in an external organization, another certificates on the letterhead of the organization is required: "Certified that his report titled.............................. is a bonafide record of the project work done by Sri/Kum............................ under any supervision and guidance, at the ....Department of.................... (Organization) towards partial fulfilment of the requirements for the award of the Degree of BSc (Computer Science) of the University of Kerala".
- References shall be IEEE format (see any IEEE magazine or transaction). Take care in use of italics and punctuation. While doing the project, keep note of all books you refer, in the correct format, and include them in alphabetical order in your reference list. Eg: A book is cited as: Kartalopoulos, S V Understanding Neural Networks and Fuzzy Logic, BPB Publishers, 1996, pp. 21-27. (pp.21-27 indicates that pages 21-27 have been referred. If the whole book is being referred, this may be omitted. If a single page is referred, say 7, it may be cited as p.**7.**
- **Report writing is NOT a hasty activity done after finishing the project**. Students must try to develop the report along with the work, so as to give it flesh and blood. Drafts should be read, modified, spell checked and grammar checked at least thrice during the course of the project and before a final printout is taken, the same may be got approved from the internal guide.
- The students should send two interim reports after the analysis and design phases of the project to internal guides. This will also help the students in their report writing.

- A soft copy of the complete documentation, including source code, should be maintained for any clarification during assessments.
- The Gantt chart, fortnightly progress reports recorded in team meeting minutes mentioned in section 3.5 should appear as appendix to the project report.

Regarding the body of the report, as an indicative example, the following is given (though students should not attempt to fit every kind of project report into this format):

- Organizational overview (of the client organization, where applicable)
- Description of the present system
- Limitations of the present system
- The Proposed system- Its advantages and features
- Context diagram of the proposed system.
- DFD of the proposed system with at least one additional level of Expansion
- Structure Chart/E-R diagrams of the System
- System flowchart
- Files or tables (for DBMS projects) list. Class names to be entered for each file in OO systems.
- List of fields or attributes (for DBMS projects) in each file or table.
- File table that shows the files/tables used by each program and the files are read, written to, updated, queried or reports were produced from them.
- Reports List with column headings and summary information for each report.
- System Coding and variable/file/table naming conventions
- System controls and standards
- Screen layouts for each data entry screen.
- Report formats for each report.

Program documentation is suggested on the following lines:

- Program id
- Program function explanation
- Program level pseudocode or flowchart.
- Data entry screen (reproduced from system documentation).
- Report layout (reproduced from system documentations)
- Decision tables, decision trees, with English Explanation where necessary.
- Program listing
- Test data
- Test results.

3.8 Methodology:

Wherever applicable, object oriented approach should be used for software development. The project report should generally contain details of the following steps (*though students should not attempt to fit every kind of project into this format*):

(a) Analysis

- Study of existing systems and its drawbacks
- Understanding the functionalities of the system in detail
- Preparation of requirements
- Conduct of Feasibility study
- Development of DFD/use case diagrams

(b) Design

- Design of each subsystems/modules
- Design of each classes
- Design of Algorithms for problem solving
- User interface /Input/ Output Design
- Any other steps if necessary

(c) Coding and Implementation

(d) Testing

(e) Security, Backup and Recovery Mechanisms

(f) On line help and User Manuals

(g) Upgradability Possibilities

3.9 Project IPR & Utilization: The intellectual property rights in all project work done by the students shall vest with the University of Kerala, except in cases where some external organizations seek undertaking from students to concede IPR in all work done in their organization or under their guidance. Where possible, students should attempt to obtain at least a joint IPR for the University. In cases where project works are of public utility, students shall be asked to publish their work including source code and documentation, in so far as their rights are clear.

4. REFERENCES

4. 1 Core

- S A Kelkar, *Software Project Management*, Prentice Hall of India
- W Alan Randolph, Barry Z. Posner, *Effective project planning and management*, PHI

4.2 Additional

- Greg Mandanis, *Software Project Management Kit for Dummies*, IDG Books Joel Henry, *Software Project management*
- Frederic P B, Mythical Man-month, *Essays on Software Engineering*, Addison Wesley David Lamport, *Latex*: *A document Preparation System*, 2/e, Pearson Edn

5. EVALUATION

5.1 Criteria for external evaluation of Major Project

External evaluation is done by an external examiner appointed by the University

The following components are to be assessed for the End Semester Evaluation of the Major Project:

● Quality of documentation	- 30 marks
● Presentation of work	- 25 marks
● Viva	- 25 marks
Total	**- 80 marks**

5.2 Criteria for internal evaluation of Major Project

Internal evaluation is be done by conducting a viva voce by a team of evaluators comprising of the concerned guides and/or Head of the Department. The following are the components for internal evaluation of the Major Project:

● Presentation of the work	-5 marks
● Individual involvement & team work/ Attendance	- 5marks
● Timely submission and assessment of 2 interim reports	-10 marks
Total	**- 20 marks**