CS010 601: Design And Analysis Of Algorithms
(Common with IT010 605)

Teaching scheme
2 hours lecture and 2 hours tutorial per week

Credits: 4

Objectives
- To develop an understanding about basic algorithms and different problem solving strategies.
- To improve creativeness and the confidence to solve non-conventional problems and expertise for analysing existing solutions.

Module I (13 hours)
Introduction and Complexity

Module II (11 hours)

Module III (14 hours)
Greedy Strategy - Control Abstraction, General Knapsack Problem, Minimum Cost Spanning Trees – PRIM’s Algorithm, Kruskal’s Algorithm, Job sequencing with deadlines.
Dynamic Programming - Principle of Optimality, Multistage Graph Problem, Forward Approach, Backward Approach, All-Pairs Shortest Paths, Traveling Salesman Problem.

Module IV (11 hours)
Backtracking – State Space Tree - Fixed Tuple and Variable Tuple Formulation - Control Abstraction – Generating Function and Bounding Function - Efficiency of the method - Monte Carlo Method – N-Queens Problem, Sum of Subsets.
Branch and Bound Techniques – FIFO, LIFO, and LC Control Abstractions, 15-puzzle.

Module V (11 hours)
Lower Bound Theory - Comparison Trees for Searching and Sorting, lower bound on comparison based algorithms, Sorting, Selection & Merging; Oracles and Adversary Arguments – Merging, Basic concepts of randomized algorithm-Las Vagas algorithm for search.

## Reference Books

CS010 602: Internet Computing

Teaching scheme
2 hours lecture and 2 hours tutorial per week

Credits: 4

Objectives

- To impart the basic concepts of Internet Computing and Java Programming
- To develop understanding about Internet Computing with the help of Java Platform and establishing network connections using Socket Programming

Module I (10 hours)

Module II (12 hours)

Module III (14hours)

Module IV (13 hours)

Module V (11 hours)
Reference Books
8) Debasish Jana, *Java and Object Oriented Programming Paradigm*, Prentice Hall of India, New Delhi, 2005
CS010 603 SYSTEM SOFTWARE

Teaching scheme

3 hours lecture and 1 hour tutorial per week

Objectives:-

To introduce the techniques adopted in the design and implementation of System Software.

Module I (12 Hrs)
Introduction:-

Macro Preprocessor

Macro Instruction Definition and Invocation. Types of Macros – Parameterised macros, Nested macros, Recursive macros. Basic functions of Macro Preprocessor – Macro expansion, Generation of unique labels. Macro preprocessor design and Algorithm - Handling conditional Macro calls, Nested Macro calls and Recursive Macro calls.[Reference (1)] Case Study : The C Preprocessor [Web- Reference (1) ]

Module - II (15 Hrs)

Assembler


Module - III (12 Hrs)

Linker and Loader

Need for Linking and Loading : The absolute loader, Program Relocation, Relocating Loader, Linking external symbols. Algorithms for the two passes of a Linking Loader.[References (2),(3)] Variants of the basic model – Automatic Library Search, Linkage Editor, Dynamic Linking. [Reference(1)] Case study : UNIX ELF and Microsoft DLL (basic structure only).

Module - IV (11 Hrs)

Text Editors : Overview of Editing, User Interface, Editor Structure. [Reference (1)]

Case Study : VI Editor (Basic ideas only)[ Reference (1)]

Debuggers : Debugging Functions and Capabilities, Relationship with other parts of the system, Debugging Methods- By Induction, Deduction and Backtracking,. [Reference (1) ,(8)] Case Study : gdb (Basic ideas only)
Module - V (10 Hrs)

**Device Driver** : Device Characteristics, Design and anatomy, Types of device driver, General Design – Character Devices and character device drivers, Block Devices and Block device drivers. *Case Study: Device Driver for the PC Speaker* [References(4), (6), (7)]

---

**REFERENCES:**

5. IBM PC Assembly Language and Programming - Peter Abel Third Edition – Prentice Hall of India.

**WEB REFERENCE:**


Note: separate subjects are provided in the syllabus in the Seventh and Fifth Semesters for the detailed discussion of the subjects marked [1] and [2] respectively.
CS010 604: Computer Networks

Teaching scheme
3 hours lecture and 1 hour tutorial per week

Credits: 4

Objectives
To develop basic knowledge on the mode of operation of different types of computer networks that are used to interconnect a distributed community of computers and various interfacing standards and protocols.

Module I (8 hours)
Network requirements, Network Architecture –layering and protocol, OSI Architecture, Internet Architecture, Performance-bandwidth and latency, Delay x bandwidth product, high speed networks.

Module II (10 hours)
Direct Link Network, Hardware Building Block, Framing-Byte Oriented Protocol, Bit Oriented Protocol, Clock Based Framing, Reliable Transmission-Stop and Wait, Sliding Window, Ethernet(802.3)-Physical properties, Access protocol, Wireless-Bluetooth, WiFi, Wimax

Module III (12 hours)
Packet Switching-Switching and Forwarding- Datagram, virtual circuit switching, Source routing Bridges and LAN Switches-Learning Bridges, Spanning tree Algorithms, Broadcast and Multicast, Limitations of bridges, Simple Internetworking-Service Model, Global Address, Datagram Forwarding in IP, address translation, Routing-network as graph, distance vector, link state, matrix

Module IV (16 hours)
End to End Protocol, Simple de-multiplexer, Reliable Byte stream, TCP-Issues, segment format, connection establishment and termination sliding window revisited, triggering transmission, adaptive retransmission, RPC-fundamentals, TCP Congestion control –additive increase, slow start, fast retransmit and fast recovery, congestion avoidance mechanism, DEC bit, Random Early Detection bit, Source Based Congestion avoidance

Module V (14 hours)
Applications -WWW, E-mail, Name Service, Network Management, Web Services Custom Application protocol, Generic Application Protocol, Overlay Networks-Peer to Peer Networks.

Reference Books
2. Introduction to data communication and networking Behrouz Forozan, TMH.
3. Computer networks, Andrew S Tanenbaum, PHI
4. Data communication, computer networks and open systems, Halsall F, Addison Wesley.
CS010 605 SOFTWARE ENGINEERING

Teaching scheme

3 hours lecture and 1 hour tutorial per week

Objectives:

To familiarize the steps in designing a Computer Software System following the conventions in Engineering Design.

To introduce the fundamentals of Structured and Object Oriented Designs and Design Tools.

Module I (12 Hrs)


Module - II (12 Hrs)

Management: Functions - Project planning - Software productivity - Productivity metrics - Cost estimation - COCOMO & COCOMO II - Project control - Work breakdown structures, Gantt charts, PERT charts - Dealing with deviations - Team organization - centralized, de-centralized, mixed - An assessment of organizations - Risk management – Configuration Management. Introduction to project management and planning CASE tools.

Module - III (12 Hrs)


Module - IV (12 Hrs)

Design activity & its objectives – Function Oriented and Object Oriented Design- Modulization techniques - module structure and its representation, interface and information hiding, categories, specific techniques to accommodate change, stepwise refinement, top-down and bottom-up design - Handling anomalies. Case Study with UML and CASE Tool support.

Module - V (12 Hrs)

References

CS010 606L01: DISTRIBUTED SYSTEMS

Teaching scheme
2 hours lecture and 2 hours tutorial per week

Credits: 4

Objectives
- To impart an introduction to distributed systems and distributed computing.
- To develop basic knowledge on distribution of data and file systems in distributed environment.
- To provide exposure to distributed database concepts.

Module I (10 hours)
Introduction to Distributed Systems, evolution, characteristics, design issues, user requirements, Distributed computing models-workstation model, workstation-server model, processor–pool model. Protocols for distributed systems -VMTP and FLIP.

Module II (12 hours)

Module III (14 hours)

Module IV (12 hours)

Module V (12 hours)
Distributed Databases: Distributed DBMS architecture, distributed query processing, transactions, concurrency control, deadlock management and Distributed Database Recovery protocols-2PC, Network Partitioning.

Reference Books
4. Andrew S Tenenbaum, Distributed Operating Systems, Pearson Education Asia
Teaching scheme
2 hours lecture and 2 hours tutorial per week

Objectives
- To impart the basic concepts of microcontrollers and their programming in assembly language and in C.
- It also focused on the 8051 microcontroller which is a widely used microcontroller.

Pre-requisites: Microprocessor systems, Advanced microprocessor and peripherals

Module I (10 hours)

Module II (12 hours)
8051 programming in C-data types and time delay – I/O programming – logical operation – data conversation program – basics of serial communication connection to RS232- serial port programming in assembly and C.

Module III (14 hours)
Basics of interrupts, interrupt sources, interrupt enable register, interrupt priority, interrupt control system, interrupt handling, single step operation, port bit latches and buffers, port structures and operation, accessing external memory.

Module IV (12 hours) Timer 0& -Timer1- T MOD SFR-mode0,mode 1,mode2,mode3-TCON SFR-serial interface-SCON SFR-mode0,mode 1,mode 2,mode3-block schematicsbaud rates-power on reset circuit-ONCE mode-on chip oscillator-external program & data memory timing diagrams.

Module V (12 hours)
PIC microcontrollers: Overview and features-PIC16C6X/7X FSR-Reset action-PIC memory organization-instructions-addressing modes.
# Reference Books

2. V Udayashankara, M S Mallikarjunaswamy, *8051 Microcontroller hardware & software application*, TMH  
5. Satish Shah, *8051 microcontrollers MCS 51 family and its variants*, Oxford higher education
CS010 606L03: User Interface Design

**Teaching scheme**

2 hours lecture and 2 hours tutorial per week

**Credits:** 4

**Objectives**
- To impart the basic concepts of User Interface Design.
- To develop understanding about human computer interaction methods that utilize more general, widespread and easier-to-learn capabilities.

**Module I (8 hours)**
Introduction: Importance of user interface – definition, importance of good design, brief history – Graphical User Interface – Web User Interface – Theories, Principles and Guidelines of User interface design

**Module II (10 hours)**
Design Process: Obstacles in development path designing for people-Understanding Human Interaction with computers, Importance of Human Characteristics, Human consideration, Human Interaction speeds – Understanding Business function

**Module III (15 hours)**

**Module IV (15 hours)**

**Module V (12 hours)**

**Reference Books**
2. Ben Shneiderman, *Designing the User Interface*, 3rd Edn., Pearson Education Asia, Delhi, 2002
5. Alan Cooper, *The Essentials of User Interface Design*, Wiley Dreamtech, Delhi, 2002
CS010 606L04 : UNIX Shell Programming
(Common with IT010 606L03)

Teaching scheme
2 hours lecture and 2 hour tutorial per week

Credits: 4

Objectives
- To provide a fair knowledge of Unix concepts and gain sharp skills in Unix Shell programming

Module 1. (8 hours)
Introduction to Unix:- Architecture of Unix, Features of Unix, Basic Unix Commands - Unix Utilities:- Introduction to unix file system, vi editor, file handling utilities, security by file permissions, process utilities, disk utilities, networking commands - Text processing utilities and backup

Module 2. (13 hours)

Module 3. (12 hours)
grep:-Operation, grep Family, Searching for File Content. sed:-Scripts, Operation, Addresses, commands, Applications, grep and sed. awk:-Execution, Fields and Records, Scripts, Operations, Patterns, Actions, Associative Arrays, String Functions, Mathematical Functions, User Defined Functions, Using System commands in awk, Applications of awk, grep and sed

Module 4. (15 hours)

Shell Programming - Korn Shell, C Shell and BASH -
Basic Script concepts, Expressions, Decisions: Making Selections, Repetition, special Parameters and Variables, changing Positional Parameters, Argument Validation, Debugging Scripts, Script Examples.

Module 5. (12 hours)
Process management:- Creation, Hierarchies, Sending signals to processes, exec, termination, Zombie, waitpid etc - Network management:- tools, Client server mechanism, address resolution, ping, telnet, ftp, dns and squid - X Window System:- Overview, Architecture, starting and stopping X, X clients and display

Syllabus - B.Tech. Information Technology
Reference Books

3. Kernighan and Pike, “Unix programming environment”, PHI. / Pearson Education
4. Graham Glass, King Ables,” Unix for programmers and users”, 3rd edition, Pearson Education
CS010 606L05: Embedded Systems

Teaching scheme
2 hours lecture and 2 hours tutorial per week

Credits: 4

Objectives
- To impart the basic concepts of Embedded System and its applications
- To develop understanding about micro controllers and programming the micro controller for the development of Embedded systems.

Module I (-12 hours)

Module II (13 hours)
Application Specific Embedded System – Domain Specific Embedded System, Designing Embedded Systems with 8bit Microcontrollers- Factors to be considered in selecting a Controller- Designing with 8051 microcontroller- 8052 microcontroller, Programming the 8051 microcontroller – Addressing modes of 8051 – the 8051 Instruction set

Module III (13 hours)

Module IV (12 hours)

Module V (10 hours)

# Reference Books

CS010 606L06: Advanced Software Environments

Teaching scheme
2 hours lecture and 2 hours tutorial per week

Credits: 4

Objectives
- To impart the basic concepts of Windows programming.
- To develop understanding about the new software environment and develop of software to meet the growing demand of the industry.

Pre-requisites: Knowledge required to study this subject (OOP concepts)

Module I (10 hours)

Module II (10 hours)

Module III (13 hours)

Module IV (13 hours)

Module V (14 hours)

Reference Books
CS010 607: Operating Systems Lab

Teaching scheme
3 hours practical per week

Credits: 2

Objectives
- To provide a practical exposure of all algorithms and behaviour of processes in the system with respect to all its timings.
- This lab also explains the allocation of process in the memory with some memory management techniques.

(Implement the following on LINUX platform. Use C for high level language implementation)

1. Basic UNIX commands

2. Shell programming
   - Command syntax
   - Write simple functions with basic tests, loops, patterns

3. Write programs using the following system calls of UNIX operating system:
   - fork, exec, getpid, exit, wait, close, stat, opendir, readdir

4. Write programs using the I/O system calls of UNIX operating system (open, read, write, etc)

5. Write C programs to simulate UNIX commands like ls, grep, etc.

6. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for FCFS and SJF. For each of the scheduling policies, compute and print the average waiting time and average turnaround time

7. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for Priority and Round robin. For each of the scheduling policies, compute and print the average waiting time and average turnaround time

8. Implement the Producer – Consumer problem using semaphores.

9. Implement inter-process communication using shared memory.

10. Implement some memory management schemes

Example for expt 10:

Free space is maintained as a linked list of nodes with each node having the starting byte address and the ending byte address of a free block. Each memory request consists of the process-id and the amount of storage space required in bytes. Allocated memory space is again maintained as a linked list of nodes with each node having the process-id, starting byte address and the ending byte address of the allocated space.

When a process finishes (taken as input) the appropriate node from the allocated list should be deleted and this free disk space should be added to the free space list. [Care should be taken to merge contiguous free blocks into one single block. This results in deleting more than one node from the free space list and changing the start and end address in the appropriate node]. For allocation use first fit, worst fit and best fit.
CS010 608 Mini Project

Teaching scheme
3 hours practical per week

Credits: 2

Objectives

- **To estimate the ability of the student in transforming the theoretical knowledge studied so far into application software.**

- **For enabling the students to gain experience in organisation and implementation of a small project and thus acquire the necessary confidence to carry out main project in the final year.**

- **To understand and gain the knowledge of software engineering practices, so as to participate and manage large software engineering projects in future.**

In this practical course, each group consisting of two/three members (four in special cases) is expected to design and develop practical solutions to real life problems related to industry, institutions and computer science research. Software life cycle should be followed during the development. The theoretical knowledge, principles and practices gained from various subjects should be applied to develop effective solutions to various computing problems. The knowledge gained during various practical subjects to work with various software tools, Designing tools, programming languages, operating systems, etc. should be utilized in various stages of development. Structured/ Object Oriented design techniques may be used for the project. Software Requirements Specification (SRS), Modeling Techniques, Design and Testing strategies should be documented properly.

A committee consisting of minimum three faculty members will perform the internal assessment of the mini project. A report on mini project should be submitted for evaluation and project work should be presented and demonstrated before the panel of examiners.

<table>
<thead>
<tr>
<th>Internal Continuous Assessment (50 marks)</th>
</tr>
</thead>
<tbody>
<tr>
<td>40% - Design and development (30% by guide and 10% by committee)</td>
</tr>
<tr>
<td>30% - Final result and Demonstration (15% by guide and 15% by committee)</td>
</tr>
<tr>
<td>20% - Report (10% by guide and 10% by committee)</td>
</tr>
<tr>
<td>10% - Regularity in the class (by guide)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>End Semester Examination (Maximum Marks-100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20% - Demonstration of mini project</td>
</tr>
<tr>
<td>50% - Practical test connected with mini project</td>
</tr>
<tr>
<td>20% - Viva voce</td>
</tr>
<tr>
<td>10% - Project report</td>
</tr>
</tbody>
</table>