Punjab Technical University, Jalandhar
Study Scheme (Batch 2012)
M.Tech System Software

Schedule of Teaching

Lecture	Tutorials	Total

Schedule of Examination

	Time (Hrs)	Theory Marks	Sessional Marks	Viva	Total
All theory Subjects	3	100	50		150
Projects			50	50	100
Seminar			100		100
Dissertation			Satisfactory/Not Satisfactory		

Semester -I

Subject Code	Name of Subject	L	T	P
SS-501	Advanced Data Structures	4	-	-
SS-503	Advanced Automata	4	-	-
SS-505	Advanced System Software Design	4	-	-
SS-507	Software Design and Construction	4	-	-
SS-509	Distributed Operating System	4	-	-
SS-511	Advanced Data Structures Lab	-	-	4
SS-513	System Software Design Lab	-	-	4

Semester -II

Subject Code	Name of Subject	L	T	P
SS-502	Research Methodology	4	-	-
SS-504	Device Driver Design and Development	4	-	-
SS-506	Mobile Application Development	4	-	-
SS- AAA	Elective-I	4	-	-
SS- BBB	Elective-II	4	-	-

Semester –III

Subject Code	Name of Subject	L	T	P
SS-CCC	Elective-III	4	-	-
SS- DDD	Elective-IV	4	-	-
SS-523	Project	-	-	-
SS-525	Seminar	-	-	-

Semester-IV

Subject Code	Name of Subject	L	T	P
SS- 500	Dissertation	-	-	-

List of Electives

Elective I SS-AAA

SS-508 Network System Design

SS-510 System Performance and Evaluation

SS-512 Security Analysis of Software

Elective II SS-BBB

SS-514 Operations Research

SS-516 Real-Time Systems and Software

SS-518 Software Architecture

Elective III SS-CCC

SS-515 Windows Kernel Programming

SS-517 Linux Kernel Programming

SS -519 Optimization Techniques

Elective IV SS-DDD

SS-520 Organization Theory and Behavior

SS-522 Total Quality Management

SS-524 Product Design and Management

SS-526 Enterprise Resource Planning

SS-501 Advanced Data Structures

L	T	P
4	-	-

Objective: This course helps students, step by step to develop algorithms and to solve real world problems. Implementing various data structures and understanding various searching & sorting techniques. To arrange data in a particular manner. These manner or set of rules is defined in the advanced data structure so that the data used in computer systems can properly used at necessary time.

Review of Elementary Data Structures: Arrays, linked lists, stacks, queues, binary trees, hashing, graphs, sorting & searching techniques.

Sparse Matrices: Properties of sparse matrices, Linked list representation of sparse matrices.

Threaded Trees: Properties of threaded trees, insertion, deletion and traversal.

AVL Trees: Properties of AVL trees, rotations, insertion and deletion.

Red-Black Trees: Properties of red-black trees, rotations, insertion and deletion.

B-Trees: Definition of B-trees, basic operations on B-trees, deleting a key from a B-tree.

Heaps: Properties of Min-max heaps, building a heap, basic operations on heaps, application of min-max heaps.

Binomial heaps: Binomial tress and binomial heaps, operations on binomial.

Fibonacci heaps: Structure of Fibonacci heaps, merge able heap operations, decreasing a key and deleting a node, bounding a maximum degree.

Data Structures for Disjoint Sets: Disjoint set operations, linked list representation of disjoint sets, disjoint set forests.

Graph Algorithms: Topological sort, minimum Spanning tree, single-source shortest paths, all-pairs shortest paths, bi-connected components, strongly connected components, cycles, articulation points, bridges.

String Matching: String-matching algorithm, Rabin-Karp algorithm, String matching with automata, Knuth-Morris-Pratt algorithm, Boyer-Moore algorithm.

NP-completeness: Complexity classes P and NP, examples of reductions.

Suggested Readings/ Books:

1. Peter Brass, "Advanced Data Structures" Cambridge University Press, 2008.
2. Kurt Mehlhorn, Peter Sanders "Algorithms and Data Structures", Springer Berlin Heidelberg, 2008.
3. A.A.Puntambekar, "Advanced Data Structures And Algorithms", Technical Publications, 2008
4. Darren Redfern, Colin Campbell, **"**Advanced Data Structures" Springer New York, 1998.
5. Frank Dehne, John Iacono, Jörg-Rüdiger Sack," Algorithms and Data Structures" 12th International Symposium, WADS 2011, NY, USA, 2011.

SS-503 Advanced Automata

L	T	P
4	-	-

Objective: At the end of this course, the students should be able to understand and explain selected advanced parts of automata theory, including parsing techniques for deterministic context-free languages, relationship between finite-state automata, automata on infinite words, and process specifications.

Finite automata and Regular Languages: Finite state systems, Deterministic, non-deterministic finite automata, equivalence of deterministic and non-deterministic finite automata, finite automates with and without €-moves, 2way finite automata with output, equivalences of mealy and Moore machines.

Properties of regular sets: The pumping lemma for regular sets, closure properties of regular sets, decision algorithm of regular sets, the Myhill-nerode theorem and minimization of finite automata.

Context free grammars: introduction to context free grammars, derivation tress, top-down and bottom up parsing methods, ambiguous context free grammers,Chomsky and greibach normal forms.

Pushdown automata: Deterministic and non-deterministic pushdown automata, equivalence of context free languages and sets accepted by pushdown automata, deterministic context free languages.

Properties of context free languages: The pumping Lemma for context free languages, closure properties of context free languages, decision algorithm for context free languages, coke-kasmi-young algorithm.

Turning machines: Introduction to turning machines, deterministic, non-deterministic, two way infinite tape, multi tape, constructions of Turing machines for n!,n*n. Post correspondence Problem, Unsolvability of halting problems.

Cellular Automata: Introduction to Cellular Automata (CA)Computing model, Neighborhood and radius, Moore and Von Neumann Architecture, advantages over conventional machine.

Undecidability: A Language That Is Not Recursively Enumerable Enumerating the Binary Strings ,Codes for Turing Machines , The Diagonalization Language , Proof That L_d Is Not Recursively Enumerable ,An Undecidable Problem That Is RE ,Recursive Languages ,Complements of Recursive and RE languages ,The Universal Language ,Undecidability of the Universal Language , Undecidable Problems About Turing Machines , Reductions Turing Machines That Accept the Empty Language ,Rice's Theorem and Properties of the RE Languages , Problems about Turing-Machine Specifications.

Intractable Problems: ,The Classes P and NP , Problems Solvable in Polynomial Time ,An Example: Kruskal's Algorithm ,Nondeterministic Polynomial Time , An NP Example: The Traveling Salesman Problem, Polynomial-Time Reductions ,NP-Complete Problems , The Satisfiability Problem ,Representing SAT Instances ,NP-Completeness of the SAT Problem.

Additional Classes of Problems : Complements of Languages in NP , The Class of Languages Co-NP , NP-Complete Problems and Co-NP , Problems Solvable in Polynomial Space Polynomial-Space Turing Machines, Relationship of PS and NPS to Previously Defined Classes , Deterministic and Nondeterministic Polynomial Space

Suggested Readings/ Books:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, "Introduction to automata theory languages and computation", Addison Wesley, 2007.
2. Peter Linz, "An Introduction to Formal Languages and Automata", Jones & Bartlett Learning, 2001.
3. Hopcroft "Introduction To Automata Theory, Languages, And Computation, 3/E", Pearson Education India, 2008.
4. Zvi Kohavi, "Finite Automata Theory 2E", Tata McGraw-Hill Education, 2007.
5. D C Kozen,"Automata and Computability", Springer,1997.
6. Luca Aceto, Monika Henzinger, Jiri Sgall,Automata, "Languages and Programming" Springer, 2011

SS-505 Advanced System Software Design

L	T	P
4	-	-

Objective: The main objective of this course is give complete knowledge about compiler design and tools used in compiler construction. At the end of course student will know that how system software is designed.

Compiler Design and Optimization: Grammars – Lexical Analysis – Syntactic Analysis – Code Generation – Heap Management – Parameter Passing Methods – Semantics of Calls and Returns – Implementing Subprograms – Stack Dynamic Local Variables – Dynamic binding of method calls to methods – Overview of Memory Management, Virtual Memory, Process Creation – Overview of I/O Systems, Device Drivers, System Boot, Symbol table structure – Local and Global Symbol table management Intermediate representation – Issues – High level, medium level, low level intermediate languages – MIR, HIR, LIR – ICAN for Intermediate code – Optimization – Early optimization – loop optimization.

Compiler Construction Tools: LEX and YACC.

Procedure optimization: In-line expansion – leaf routine optimization and shrink wrapping – register allocation and assignment – graph coloring – data flow analysis – constant propagation – alias analysis – register allocation – global references – Optimization for memory hierarchy - Code Scheduling – Instruction scheduling – Speculative scheduling – Software pipelining –trace scheduling – Run-time support – Register usage – local stack frame – run-time stack – Code sharing – position–independent code.

Introduction to Virtual Machines (VM): Pascal P-Code VM – Object-Oriented VMs – Java VM Architecture – Common Language Infrastructure – Dynamic Class Loading – Security – Garbage Collection – Optimization.

Emulation: Interpretation and Binary Translation – Instruction Set Issues – Process Virtual Machines – Profiling – Migration – Grids – Examples of real world implementations of system software.

Suggested Readings/ Books:

1. Steven S. Muchnick, "Advanced Compiler Design Implementation", Morgan Koffman – Elsevier Science, India, First Edition 2004
2. James E Smith and Ravi Nair, "Virtual Machines", Elsevier, 2005.
3. Robert W. Sebesta , "Concepts of Programming Languages", 7th ed., Pearson Education, 2006.
4. Aho, "Compilers: Principles, Techniques and Tools", Pearson Education India, 1999.
5. Iain D Craig, "Virtual Machines", Springer, 2006.

SS-507 Software Design and Construction

L	T	P
4	-	-

Objective: "Software Design and Construction" aims to develop good practice in software engineering through individual, small group, and large group programming projects. Students will gain experience with good programming and project management techniques through the design, implementation, testing, and documentation of software applications.

Software Design: Design concepts, the design model, software architecture, architectural design, data design, component level design, and user interface design.
Rational Software Architect: Use case diagram: Requirement Capture with Use case, Building blocks of Use Case diagram - actors, use case guidelines for use case models, Relationships between use cases - extend, include, generalize. Activity diagram :Elements of Activity Diagram - Action state, Activity state, Object node, Control and Object flow, Transition (Fork, Merge, Join) , Guidelines for Creating Activity Diagrams,Activity Diagram - Action Decomposition (Rake)., Partition - Swim Lane.

Software Construction: Basics of object-oriented approach, object-oriented programming and languages, Scope of class members-public, private, protected. Class constructor, destructor, copy constructor, virtual destructor. Derived classes, scope of derivation-public, private, protected. Virtual functions, Function overloading. Friend functions and friend classes, Operator overloading, Dynamic memory allocation to classes and class members, new and delete operators. Overloading new and delete operators. Explicit type conversion operators. Input output streams, Stream class hierarchies, standard I/O objects: cin, cout, cerr, overloading <<, >> operators, File Streams, opening, reading, writing to file. File pointers and their manipulators,Introduction to templates and container classes.

TEST MANAGEMENT AND AUTOMATION: Introduction – Test Planning – Test Management –Software test automation – Scope of automation – Test automation tools – Generic requirement for test tool/framework – Selecting a test tool – Challenges in automation.

Suggested Readings/ Books:

1. Grady Booch, James Rumbaugh, Ivar Jacobson, "The Unified Modeling Language User", Addison Wesley Professional Guide, Pearson Education,2005.
2. James Rumbaugh, Ivar Jacobson, Grady Booch,"The Unified Modeling Language Reference Manual", Addison-Wesley, New York
3. Grady Booch, "Object-Oriented Analysis and Design", Pearson Education , 2002.
4. Roger S. Pressman,"Software Engineering, A Practitioner's Approach", McGrawHill International Edition, 2009.
5. J. Rumbaugh, "Object-Oriented Modeling and Design", Prentice Hall, 2004.
6. G. Schneider, Applying Use Cases: A Practical Guide: Addison-Wesley Object Technology Series, Addison-Wesley, 2001.
7. Marget A.Eills and Bjame Stroustrup, "The Annotated C++ Reference Manual", Addison Wesley, 1990.
8. Terry Quatrani, Jim Palistrant, "Visual Modeling With IBM Rational Software Architect And UML", Prentice Hall Professional, 2006.
9. Vesta Bagheri Satari, "Model Transformation with Rational Software Architect", V.B. Satari, 2008.
10. Glenford J Myers, Corey Sandler, Tom Badgett and Todd M Thomas, "The Art of Software Testing", Wiley, USA, 2004.

SS-509 Distributed Operating System

L	T	P
4	-	-

Objective: The main principle of this course is to processes, communication, naming, synchronization, replication and consistency.

Overview of Operating System: Operating System - concept, need and requirements of operating system, Processor, Memory, Device and File management, Virtual memory, Pipes, Deadlocks and Protection issues, Comparative study of Various/ types of operating systems.

Introduction to Distributed system: Goal, Hardware Concepts, Software concepts, Design issues.

Communication in distributed system: Layered protocols, client server model, remote procedure call, group communication, Comparison of Client Server Vs. Distributed operating system.

Synchronization in distributed system: Clock synchronization, mutual exclusion, election algorithms, automatic transaction, deadlocks in distributed systems.

Processes and processors in distributed systems: Threads, System models, Processor allocation, Scheduling in distributed systems.

Distributed file system: Distributed file system, Design and Implementation trends in distributed file system.

Case study: Distributed File System: Linux, Windows, AIX.

Suggested Readings/ Books:

1. International Business Machines Corporation, "AIX operating system commands reference" International Business Machines Corp., 2008.
2. Andrew S. Tanenbaum, "Distributed Operating Systems" Pearson Education India, 2005.
3. Pradeep Kumar Sinha, "Distributed operating systems: concepts and design", IEEE Press, 1997.
4. Doreen L. Galli "Distributed operating systems: concepts and practice", Prentice Hall, 2000.

SS-511 Advanced Data Structure Lab

L	T	P
-	-	4

The Students are required to implement the applications based on SS-501

SS-513 System Software Design Lab

L	T	P
-	-	4

The Students are required to implement the applications based on SS-505.

SS-502 Research Methodology

L	T	P
4	-	-

Objective: Provides in depth knowledge about the systematic process of collecting and analyzing information (data) in order to increase our understanding of the phenomenon with which we are concerned or interested.

Nature and Objectives of research: Methods of research: historical, descriptive and experimental. Study and formulation of research problem. Scope of research and formulation of hypotheses; Feasibility, preparation and presentation of research proposal.

Introduction to statistical analysis: Measures of central tendency and dispersion: mean, median, mode, range, mean deviation and standard deviation.

Regression and correlation analysis: Probability and probability distributions; Binomial, Poisson, Geometric, Negative binomial, Uniform, Exponential, Normal and Log-normal distribution. Basic ideas of testing of hypotheses; Tests of significance based on normal, t and Chi-square distributions. Analysis of variance technique.

Design of experiments: Basic principles, study of completely randomized and randomized block designs. Edition and tabulation of results, presentation of results using figures, tables and text, quoting of references and preparing bibliography.

Suggested Readings/ Books:

1. Levin, Levin Richard , "Statistics for Management", Pearson Education India, 2011.
2. N.K Malhotra, "Marketing Research An Applied Orientation", Pearson Education India, 2010.
3. W.G Zikmund, "Business Research Methods", Thomson/South-Western, 2003.
4. K.N Krishnaswami , Sivakumar, A. I. and Mathirajan, M., "Management Research Methodology", Pearson Education India, 2009 .
5. Kothari, "Research Methodology Methods and techniques", New Age International, 2009.
6. Paul L. Meyer, "Introduction to Probability & Statistical, Applications: Oxford", Addison-Wesley Pub. Co., 2007.

SS-504 Device Driver Design and Development

L	T	P
4	-	-

Objective: This course provides substantial practice with the key steps in developing device drivers. The course shows that how device drivers work with the kernel, how to compile and load drivers.

Windows Driver Foundation (WDF): Introduction to the Windows Driver Foundation, Architecture of the WDF, I/O Flow and Dispatching in WDF Drivers, Plug and Play and Power Management in WDF Drivers, Writing USB Drivers with WDF.

Kernel-Mode Driver Framework (KMDF): Architecture of the Kernel-Mode Driver Framework, DMA Support in KMDF Drivers, An Introduction to How to Build, Install, Test, and Debug KMDF Drivers,Frameworks Verifier, KMDF Log.

User-Mode Driver Framework (UMDF): Introduction to the UMDF, Architecture, An Introduction to Component Object Model for UMDF Developers.

General Windows Driver Concepts: Kernel-Mode Fundamentals: Common Driver Reliability Issues Scheduling, Thread Context, and request level IRQL, Locks, Deadlocks, and Synchronization Security Topics: Threat Modeling for Drivers.

Tools for Driver Developers: Driver Installation, Debugging and Tracing, Static Driver Verifier, PRE*f*ast for Drivers

Suggested Readings/ Books:

1. Ronald D. Reeves Ph.D., "Windows 7 Device Driver", Pearson Education, 2010.
2. Arthur, Baker, Jerry Lozano, "The Windows 2000 Device Driver Book: A Guide for Programmers", Prentice Hall Professional, 2001.
3. Chris Cant, "Writing Windows WDM Device Drivers", R&D Books, 1999.
4. Penny Orwick, Guy Smith "Developing Drivers with the Windows® Driver Foundation", O'Reilly Media, Inc., 2010.

SS-506 Mobile Application Development

L	T	P
4	-	-

Objective: Mobile Application Development provides information to allow students to write, test and deploy applications into the target platform environment. Mobile application development is the process by which application software is developed for low-power handheld devices, such as personal digital assistants, enterprise digital assistants or mobile.

Introduction: Mobile operating system, Operating system structure, Constraints and Restrictions, Hardware configuration with mobile operating system, Features: Multitasking Scheduling, Memory Allocation, File System Interface, Keypad Interface, I/O Interface, Protection and Security, Multimedia features.

Windows Phone: Introduction to Windows Phone, Architecture-framework, Cloud Services, Memory management-application memory management, Application development cycle, Deployment of application.

Android: Introduction to Android, Architecture-dalvik, Memory management-page based, Application development methods-agile, waterfall, Deployment-signing process.

iOS: Introduction to iOS, Architecture-cocoa touch layer, Memory management-manual retain release, automatic reference counting, Application development methods- iOS sdk.

Case Study: Design and development of Application using mobile application development platforms e.g. IBM WorkLight, Eclipse, Xcode, XPages.

Suggested Readings/ Books:

1. Jeff McWherter, Scott Gowell "Professional Mobile Application Development", John Wiley & Sons, 2012.
2. Jennifer Kyrnin, "Sams Teach Yourself HTML5 Mobile Application Development in 24 Hours", Sams Publishing, 2011.
3. Damon Oehlman, Sébastien Blanc, "Pro Android Web Apps: Develop for Android using HTML5, CSS3 & JavaScript", Apress, 2011.
4. Burd, "Android Application Development All-in-One For Dummies", John Wiley & Sons, 2011.
5. Henry Lee, Eugene Chuvyrov, "Beginning Windows Phone App Development", Apress, 2012.
6. Neal Goldstein, Tony Bove, "iPhone Application Development All-In-One For Dummies", John Wiley & Sons, 2010.

Elective -I SS-AAA

SS-508 Network System Design

L	T	P
4	-	-

Objective: This course helps student a quick review on packet header formats, Traditional Protocol Processing Systems; Network Processors. The scope of this course includes the concepts, principles, and hardware and software architectures that are the underpinnings of the design and implementation of network.

Introduction: Review of Protocols & Packet Format, Network Systems & the Internet, Network Systems Engineering, Packet Processing, Achieving high speed, Network Speed, Hardware, Software & hybrids. A conventional computer system, Fetch-Store paradigm, Network Interface Card functionality, onboard addresses recognition, Packet Buffering, Promiscuous mode.

Protocols: IP Datagram, Fragmentation, Reassembly, Forwarding, TCP Splicing.

Network Processors: RISC vs CISC, Network Processors, Ingress & Egress Processing, Parallel & Distributed Architecture, Network Processor Design, Examples of Commercial Network Processors, Overview of Intel Network Processor, Micro engine Programming, Core Programming.

Suggested Readings/ Books:

1. Douglas Comer, "Network Systems Design using Network Processor", Pearson Education, 2004.
2. Erik J. Johnson, Aaron Kunze, "IXP 1200 Programming", Intel Press, 2002.
3. Sumit Ghosh, "Principles of Secure Network Systems Design", Springer, 2002.
4. Stephen Northcutt, Mark T. Edmead, "Inside network perimeter security: the definitive guide to firewalls, VPNs, routers, and intrusion detection systems", New Riders, 2003.

SS-510 System Performance and Evaluation

L	T	P
4	-	-

Objective: This course is providing to learn techniques for modeling computing systems for performance evaluation through analytical as well as simulation methods. The students would learn how to model computing systems and network communication protocols to evaluate their performance under different operating conditions.

The art of performance evaluation: Professional organizations, journals, and conferences, Performance Projects, Common Mistakes in Performance Evaluation, A systematic approach to Performance Evaluation, Selection of techniques – Performance metrics, Utility classification, setting performance requirements.

Types of workloads: Instruction mixes, Kernels, Synthetic Programs, Application Benchmarks, Art of Workload selection, services exercised, level of detail, Representativeness, Timeliness, Other considerations in Workload selection, Workload Characterization Techniques, Terminology, Averaging, Specifying Dispersion , Single-Parameter and Multi parameter Histograms, Principal-Component Analysis, Markov models, Clustering.

Monitors – Terminology: Classifications, Software and Hardware Monitors, Firmware and Hybrid Monitors, Distributed-System Monitors, Program Execution Monitors, Accounting Logs, Analysis and Interpretation of log data, Capacity Planning and Benchmarking, Load Drivers, Remote-Terminal Emulation, Art of Data Representation, Guidelines for preparing good graphical charts, Gantt Charts, Kiviat Charts, Schumacher Charts.

Summarizing Measured Data: Basic Probability and Statistics Concepts, Geometric Mean, Harmonic Mean, Mean of a Ratio, Index of Dispersion, Determining Distribution of Data -Sample versus Population, Confidence Interval for the Mean, Testing for a Zero Mean, Hypothesis Testing versus Confidence Intervals and levels, Confidence Intervals for Proportions, Determining Sample Size.

Suggested Readings/ Books:

1. R.K Jain, "The Art of Computer Systems Performance Analysis – Techniques for Experimental Design, Measurement, Simulation, and Modeling", Wiley-India, 2008.
2. Daniel, Menascé, Virgílio, Almeida, Lawrence, "System Performance Evaluation: Methodologies and Applications", Prentice Hall Professional, 2004.
3. Krishna Kant, M. M. Srinivasan, "Introduction to computer system performance evaluation", McGraw-Hill, 1992.

SS-512 Security Analysis of Software

L	T	P
4	-	-

Objective: This Course presents a clear and useful introduction to the art and science of information security. It also insights and realistic examples will help student understand the crucial links between security theory and the day-to-day security challenges of IT environments. The principles underlying both policies and mechanisms, and how attackers can subvert these tools--as well as how to defend against attackers.

Overview of Computer Security: Threats, risks, vulnerabilities, safeguards, attacks, exploits, Information states, Security at the various states of information- processing, storage and transmission; Definition of security based on current state and reachable states, Comprehensive model of security, Confidentiality, integrity and availability, Risk management, corrective action, risk assessment and physical security.

Access Control: Access control matrix, Access control lists, Capabilities, Role-based access, control and Application dependence.

Security Policies: Types of policies, Role of trust, Information states and procedures, Types of access control, Separation of duties, Application dependence, Importance for automated information systems (AIS) and Security planning Confidentiality Policies - Goals and definitions, Bell-LaPadula model and Multi-level security. Integrity Policies - Goals and definitions, Information states and procedures, Operating system integrity, Biba model and Clark-Wilson model Hybrid Policies - Chinese Wall model and Role-Based Access Control

Administrative policies: Purposes, Back-up policies, E-mail security and privacy policies, Wireless policies, Internet security policies, Incident response policies, Testing and validation policies, Application development control.

Suggested Readings/ Books:

1. Matt Bishop, "Introduction to Computer Security", Addison Wesley, 2005.
2. William Stallings, “Network Security Essentials: Applications and Standards”, Prentice Hall, 2007.
3. Ferguson, “Safety and Security”, Infobase Publishing, 2007
4. Viega, Gary McGraw, “Building Secure Software: How to Avoid Security Problems the Right Way”, Addison-Wesley, Boston, 2002.

ELECTIVE-II SS-BBB

SS-514 Operations Research

L	T	P
4	-	-

Objective: This course deals with the theoretical aspects of operations research, and explains the concepts with practical examples. It begins by focusing on the need and prerequisites of operations research and moves on to discuss topics such as linear programming, integer programming, nonlinear programming, assignment problems, and inventory models in sufficient detail. This course helps students how to achieve different goals in the order of priority to optimize the objective function, various criteria of decision making under certainty, uncertainty and risk.

Introduction: Development, Definition, Characteristics and Phases, Types of models, operation Research models and their applications.

Allocation: Linear Programming Problem Formulation, Graphical solution, Simplex method, Artificial variables techniques: Two–phase method, Big-M method, Duality Principle.

Transportation Problem: Formulation, Optimal solution, unbalanced transportation problem, Degeneracy. Assignment problem, Formulation, Optimal solution, Variants of Assignment Problem, Traveling Salesman problem.

Sequencing: Introduction, Flow, Shop sequencing, n jobs through two machines, n jobs through three machines, Job shop sequencing, two jobs through 'm' machines.

Replacement: Introduction, Replacement of items that deteriorate with time, when money value is not counted and counted, Replacement of items that fail completely, group replacement.

Theory of Games: Introduction, Minimax (maximin), Criterion and optimal strategy, Solution of games with saddle points, Rectangular games without saddle points, 2 X 2 games, dominance principle, m X 2 & 2 X n games, and graphical method.

Waiting Lines : Introduction, Single Channel, Poisson arrivals, exponential service times, with infinite population and finite population models, Multichannel, Poisson arrivals, exponential service times with infinite population single channel Poisson arrivals.

Inventory : Introduction, Single item, Deterministic models, Purchase inventory models with one price break and multiple price breaks, shortages are not allowed, Stochastic models, demand may be discrete variable or continuous variable, Instantaneous production, Instantaneous demand and continuous demand and no set up cost.

Dynamic Programming: Introduction, Bellman's Principle of optimality, Applications of dynamic programming, capital budgeting problem, shortest path problem, linear programming problem.

Simulation: Definition, Types of simulation models, phases of simulation, applications of simulation, Inventory and Queuing problems, Advantages and Disadvantages, Simulation Languages.

Suggested Readings/ Books:

1. A. M. Natarajan, P. Balasubramani,"Operations Research",Pearson Education India, 2006.
2. Shah Gor & Soni, "Operations Research" PHI Learning Pvt. Ltd., 2007.
3. Panneerselvam, "Operations Research" PHI Learning Pvt. Ltd., 2006.
4. Sen, "Operations Research: Algorithms and Applications", PHI Learning Pvt. Ltd., 2010.

SS-516 Real-Time Systems and Software

L	T	P
4	-	-

Objective: The concepts and principles, of this course provide students with an accessible approach to software design. It presents several examples of commercial and research systems throughout the steps to explain and justify the concepts. Real-Time System Software presented technically diverse, including discussions of state machines, logic, concurrent programming, and scheduling algorithms

Introduction: RTS, Characteristics of RTS, RTOS, Types of RTOS, Characteristics of RTOS, Processors and micro controllers of RTS, Skill set required for various types of RTS.

S/W Engg Involved: SDLC for RTS, Process models for RTS-SPIRAL, incremental Xtreame, prototyping, RAD, Risk & Failure Analysis.

Requirement Analysis: RT requirement eliciation and analysis using structured and object-oriented approach, Applications of formal methods for requirement specification.

Architecture & Design: Architecture properties, RT Architecture, design temporal & non temporal,. Techniques, scheduling- (Tasks, T&S, RM scheduling).

Testing of RTS: verification& validation, test strategy, RTS test techniques.

Languages& Tools: Introduction to languages used for development of RTS, Introduction to tools-Rational Test Real Time, STATE-MATE from i-Logix, Software through Pictures (StP).

Suggested Readings/ Books:

1. Alan C Shaw, "Real-Time Systems and software", John Wiely and Sons, 2001.
2. Philip Laplante, "Real-Time Systems and design and analysis", IEEE computer society press,2004 .
3. J. E .Cooling, "Software design for Real-Time Systems", Chapman and Hall, 1991.
4. Krishna M Kavi, "Real-Time Systems: abstraction, languages and design methodologies", IEEE Computer Society press,1998.

SS-518 Software Architecture

L	T	P
4	-	-

Objective: This course defining a structured solution that meets all of the technical and operational requirements, while optimizing common quality attributes such as performance, security, and manageability. It involves a series of decisions based on a wide range of factors, and each of these decisions can have considerable impact on the quality, performance, maintainability, and overall success of the application.

Basics of Software Architecture: Architecture Business Cycle, Architecture Patterns, Reference Model and Reference Architecture, Architecture Structure and Views, Product Line Architecture, Functional and Non-functional Properties of Software Architectures.

Enabling Techniques for Software Architecture: Coupling and Cohesion, Sufficiency, Completeness and Primitiveness, Separation of Policy and Implementation, Separation of Interface and Implementation.

Architectural Styles: Pipes and Filters, Data Abstraction and Object-Orientation, Event-Based, Implicit Invocation, Layered Systems, Repositories, Interpreters, Process Control, Heterogeneous Architectures, Case studies based on architectural styles.

Understanding and Achieving Software Qualities: Changeability, Efficiency, Interoperability, Reliability, Testability, Reusability, Security, Usability, Fault tolerant software, Tactics to achieve software qualities.

Designing of Software Architecture: Function Oriented Design, Object Oriented Design, Attribute Driven Design of Software Architecture, Case Studies.

Documenting Software Architecture: Software Architecture Documentation Template, Use of Documentation, Creation of different views of Software Architecture with UML.

Reconstructing Software Architecture: Phases of Reconstruction, Uses of Reconstruction, Reconstruction of Software Architecture using tool.

Suggested Readings/ Books:

1. Bass Len, Clements Paul, Kazman Rick, “Software Architecture in Practice”, dorling Kingsley, 2012.
2. Jan Bosch, “Software Architecture: System Design, Development and Maintenance”, Springer, 2002.
3. Oliver Vogel, “Software Architecture: A Comprehensive Framework and Guide for Practitioners”, Springer, 2011.
4. Shaw M, Garlan D, “Software Architecture Perspectives on an Emerging Discipline”, Prentice-Hall, 1996.
5. Booch G., Rumbaugh J., Jacobson I., “The Unified Modeling Language User Guide”, Addison-Wesley, 2005.

ELECTIVE-III SS-CCC

SS-515 Windows Kernel Programming

L	T	P
4	-	-

Objective: This course has been designed to provide students with the necessary skills and knowledge required to develop, debug and troubleshoot low-level Windows kernel. The course provides a detailed view of the Windows operating system architecture.

Windows Operating System Architecture Overview: A brief review of the general architecture of the Windows operating system.

Kernel Mode and Key Structures: Fundamentals of Kernel Mode in Windows, including coverage of key object types (Dispatcher, Control, Executive) and data structures (KPCR, KTHREAD, EPROCESS) that Windows uses.

Device Stacks – The Windows I/O Subsystem: Working of PnP Manager, The role of Function Drivers, Bus Drivers, and Filter Drivers is discussed. PDOs, FDOs, and Filter Device Objects

Installing Legacy Drivers on Windows: How to create installation control files for "legacy" style, software-only drivers. PnP (WDM) driver installation using INF files.

Building and Debugging: WDM drivers, how to setup and use the Windows kernel mode debugger, WinDbg.

Interrupt Request Levels and Deferred Procedure Calls: Windows synchronizes kernel mode activity by using a set of Interrupt Request Levels (IRQLs), how IRQLs are used to achieve synchronization within the OS, processing that occurs at these IRQLs - including Deferred Procedure Calls (DPCs), dispatching.

Passing Requests to Other Drivers: IRPs, I/O Stack Locations, Synchronous and asynchronous IRP completion, completion routines.

Other Kernel Extensions: Windows operating system using software only drivers, Object Manager, Process Manager, and Registry callbacks.

Suggested Readings/ Books:

1. Lee Holmes, “Windows PowerShell Cookbook: The Complete Guide to Scripting Microsoft's Command Shel”, O'Reilly Media, Inc., 2013.
2. Walter Oney, “Programming the Microsoft® Windows® Driver Model”, O'Reilly Media, Inc., 2010.
3. Arthur H. Baker, Jerry Lozano, “The Windows 2000 Device Driver Book: A Guide for Programmers”, Prentice Hall Professional, 2001.

SS-517 Linux Kernel Programming

L	T	P
4	-	-

Objective: Linux Kernel Development details the design and implementation of the Linux kernel, it helps student to write and develop kernel code. Process management, scheduling, time management and timers, system call interface, memory addressing and management, caching layers, VFS, kernel synchronization, debugging, and the kernel community included in this course.

Introduction to the Linux Kernel: Overview of Operating Systems and Kernels, Linux Versus Classic Unix Kernels, The Kernel Source Tree, Building the Kernel**.**

Process Management: Process Descriptor and the Task Structure**,** Process Creation**,** The Linux Implementation of Threads**,** Process Termination**,** Process Scheduling: The Linux Scheduling Algorithm**,** Preemption and Context Switching**,** Real-Time.

System Calls: APIs, POSIX, and the C Library, Syscalls, System Call Handler, System Call Implementation.

Interrupts and Halves: Interrupts and Interrupt Handlers, Bottom Halves and Deferring Work.

Synchronization and Memory Management: Kernel Synchronization Methods, Timers and Time Management, The Virtual Filesystem, the Block I/O Layer, The Process Address Space. The Page Cache and Page Writeback, Modules, kobjects and sysfs. Debugging, Portability, Patches, Hacking, and the Community**.**

Suggested Readings/ Books:

1. Love, "Linux Kernel Development", Pearson Education India, 2005.
2. William Stallings, "Operating Systems: Internals and Design Principles", Test, 2008.
3. Andrew S. Tanenbaum "Operating systems: design and implementation" Prentice-Hall, 2008.

SS-521 Optimization Techniques

L	T	P
4	-	-

Objective: Optimization Techniques provides a number of optimization algorithms, simple single-variable optimization techniques, and also give unconstrained and constrained optimization techniques.

Introduction: Engineering applications of optimization. Design variables, Constraints, Objectives function, Variable bounds, Statement and formulation of an optimization problem, Optimization problems, Classification of optimization problems, Different optimization algorithms.

Optimal Point: Local optimal point, Global optimal point and Inflection point.

Single Variable Optimization Techniques:· Optimality criterion, Bracketing method (Bounding phase method), Region elimination methods (Internal halving method, Golden section search method), Point estimation method (successive quadratic estimation methods)Gradient-based methods (Newton-Raphson method, Bisection method, secant. Cubic search method), Root finding using optimization techniques.

Multivariable Optimization Techniques: Optimality criterion, Unidirectional search method, Direct Search method (Hooke-Jeeves Pattern Search method, Powell's conjugate direction method) Gradient-based methods (Steepest descent method, Newton's method, Marquardt's methods)

Constrained Optimization Algorithms: Kuhn-Tucker conditions, Transformation method (Penalty function method), Direct search for constrained minimization (variable elimination method, complex search method)

Linear Programming: Linear programming problems, Simplex method of linear programming techniques.

Suggested Readings/ Books:

1. Kalyanmoy Deb, "Optimization for Engineering Design: Algorithms and Examples", PHI Learning Pvt. Ltd., 2004.
2. S. S. Rao, "Engineering Optimization: Theory and Practice", New Age International, 1996.
3. Willis Harmon Ray, Julian Szekely "Process optimization, with applications in metallurgy and chemical engineering" Wiley, 2007.
4. Gordon, Beveridge, Robert Samuel Schechter "Optimization: theory and practice", McGraw-Hill, 2010.

ELECTIVE-IV SS-DDD

SS-520 Organization Theory and Behavior

L	T	P
4	-	-

Objective: Understanding the meaning of Organizational behavior, fundamental concepts connected with Organizational behavior, basic approaches of Organizational behavior, framing the study of Organizational behavior, goals of organizational behavior, Knowing the importance of Organizational behavior for the managers.

Concept of organization and management: Development of management thought, different theories of management, Japanese management. Planning and planning process, Decision making. MBO, Decentralization, Span of management, Delegation. Line staff and functional relationship. Beurocratic organization.

Role of behavioral sciences in organization: Individual behaviour, different theories of motivation. Interpersonal and group behaviour, transactional analysis and group dynamics. Importance of human relations. Controlling and directing human behaviour in organization, Leadership, theories of leadership and leadership styles, managerial grid, organizational conflicts, organizational effectiveness, Communication significance, process and variables.

Business Cycles: Nature and phases of a business cycle, Theories of business cycles-psychological, profit, monetary, innovation, cobweb, Samuelson and Hicks Theories, Inflation, Definition, characteristics and types, Inflation in terms of demand – pull and cost-push factors, Effects of inflation.

Concept of personal management and industrial relations: role and scope, Planning personnel functions – Human resource development, functions and operations of personnel department, employees selection, recruitment and training, Job description and analysis, career planning, transfers and promotions, Compensation planning, wages and salary administration, Concept of workers participation in management.

Suggested Readings/ Books:

1. John W. Newstrom "Organizational Behavior: Human Behaviorat Work", McGraw-Hill/Irwin, 2010.
2. Michael R. Baye, "Managerial Economics and Business Strategy", McGraw-Hill/Irwin, 2010.
3. Nirmal Singh, "Organisational Behaviour: Concepts, Theory and Practices : Managing People and Organisations in the 21st Century", Deep and Deep Publications, 2001.
4. R S Dwivedi, "Human Relations And Organisational Behaviour", Macmillan, 2001.
5. S K Srivastava, "Organization Behaviour And Management", Sarup & Sons, 2005.

SS-522 Total Quality Management

L	T	P
4	-	-

Objective: This course is designed to give students fundamentals of Total Quality, Management with emphasis on contemporary quality planning, control and management approaches, implementations and criticisms.

Introduction: Quality – Basic concepts, dimensions, economics of quality, quality Gurus,TQM: Definition, evolution, journey from inspection to TQM, comparison at different stages, dimensions of TQM, TQM viewpoints, reasons for adopting TQM.

Introspection to TQM environment: Sphere of TQM, components of TQM, TQM – Managing Total Quality, Factors affecting TQM environment, Classification and interaction among factors, Researchers' viewpoint, TQM as a system, steps in TQM implementation, Roadblocks in TQM implementation, Reasons for TQM failure.

Role of soft options in TQM: Hard vs. Soft factors, Role and expectation of employer, employee, customer and supplier from organization and vice versa. Human factors in TQM, Role of top management commitment, work culture, motivation, coordination, attitude, innovation.

Quality initiatives in organizations: Role of tools and techniques in TQM, Classification of tools and techniques – Problem identification, Data analysis, Graphical, Creativity, Companywide.
Brief description of Quality awards – MBNQA, Deming award, European quality award, Australian quality award.

TQM Effectiveness: Impact of TQM, Need and difficulty in measuring TQM effect, Parameters governing effect of TQM and the attributes thereof.

Suggested Readings/ Books:

1. Besterfield, "Total Quality Management", Pearson Education India, 2011.
2. Logothetis, "Managing For Total Quality: From Deming To Taguchi And SPC", PHI, 2002.
3. Armand Vallin Feigenbaum, "Total quality control", McGraw-Hill, 2007.
4. John S. Oakland, Peter Morris, "TQM: A Pictorial Guide for Managers", Butterworth-Heinemann Limited, 1997.

SS-524 Product Design and Management

L	T	P
4	-	-

Objective: At the completion of this course, the student should be able to examine the design and performance of supply networks and processes in different business contexts. Students develop capabilities in logistics, digital coordination for supply chain integration, inventory management; risk pooling, procurement, product and process.

Introduction: Characteristics of successful product development, Design and development of products, duration and cost of product development, the challenges of product development.

Development Processes and Organizations: A generic development process, concept development: the front-end process, adopting the generic product development process, the AMF development process, product development organizations, the AMF organization.

Product Planning: The product planning process, identify opportunities. Evaluate and prioritize projects, allocate resources and plan timing, complete pre project planning, reflect all the results and the process.

Identifying Customer Needs: Gather raw data from customers, interpret raw data in terms of customer needs, organize the needs into a hierarchy, establish the relative importance of the needs and reflect on the results and the process.

Product Specifications: What are specifications, when are specifications established, establishing target specifications, setting the final specifications.

Concept Generation: The activity of concept generation clarify the problem, search externally, search internally, explore systematically, reflect on the results and the process.

Concept Selection: Overview of methodology, concept screening, and concept scoring,

Concept Testing: Define the purpose of concept test, choose a survey population, choose a survey format, communicate the concept, measure customer response, interpret the result, reflect on the results and the process.

Product Architecture: What is product architecture, implications of the architecture, establishing the architecture, variety and supply chain considerations, platform planning, related system level design issues.

Industrial Design: Assessing the need for industrial design, the impact of industrial design, industrial design process, managing the industrial design process, assessing the quality of industrial design.

Design for Manufacturing: Definition, estimation of manufacturing cost, reducing the cost of components, assembly, supporting production, impact of DFM on other factors.

Prototyping: Prototyping basics, principles of prototyping, technologies, planning for prototypes.

Product Development Economics: Elements of economic analysis, base case financial mode,. Sensitive analysis, project trade-offs, influence of qualitative factors on project success, qualitative analysis.

Managing Projects: Understanding and representing task, baseline project planning, accelerating projects, project execution, postmortem project evaluation.

Suggested Readings/ Books:

1. A C Chitale, R C Gupta, "Product Design and Manufacturing", PHI Learning Pvt. Ltd., 2007.
2. Ulrich, Steven D. Eppinger, "Product Design and Development", Tata McGraw-Hill Education, 2003.
3. Butterworth Heinmann, "New Product Development", Oxford UCI, 1997.
4. Geoffery Boothroyd, Peter Dewhurst and Winston Knight, "Product Design for Manufacture and Assembly", CRC Press, 2010.

SS-526 Enterprise Resource Planning

L	T	P
4	-	-

Objective: The ERP will develop and implement curricula in the students that bring ERP concepts and business disciplines hands-on exposure to how enterprise-wide information systems support the planning and management of business processes. It also provides the study about the connection between all the business disciplines in the real world and how ERP systems support the planning and management of business processes.

Introduction to ERP: An Overview, Enterprise – An Overview, Benefits of ERP, ERP and Related Technologies, Business Process Reengineering (BPR), Data Warehousing, Data Mining, OLAP, SCM.

ERP Implementation: Lifecycle, Implementation Methodology, Hidden Costs, Organizing the Implementation, Vendors, Consultants and Users, Contracts with Vendors, Consultants and Employees, Project Management and Monitoring.

Business modules: ERP Package, Finance, Manufacturing, Human Resources, Plant Maintenance, Materials Management, Quality Management, Sales and Distribution.

ERP Market: SAP AG, Peoplesoft, Baan, JD Edwards, Oracle, QAD, SSA.

Present and Future: Turbo Charge the ERP System, EIA, ERP and e-Commerce, ERP and Internet, Future Directions.

Suggested Readings/ Books:

1. Joseph A Brady, Ellen F Monk, Bret Wagner, "Concepts in Enterprise Resource Planning", Thompson Course Technology, USA, 2001.
2. Vinod Kumar Garg and Venkitakrishnan N K, "Enterprise Resource Planning – Concepts and Practice", PHI, New Delhi, 2003
3. Alexis Leon, "ERP Demystified", Tata McGraw Hill, New Delhi, 2000.
4. Ravi Shankar, S. Jaiswal, "Enterprise Resource Planning", Galgotia, 1999.