| | | | |
|---|---|---|---|
| **Course Title** | Numerical Methods, Operations Research & Statistics | **Semester** | IV |
| **Course Code** | MVJ19MIS41 | **CIE** | 50 |
| **Total No. of Contact Hours** | 60 L : T : P :: 40 : 0 : 20 | **SEE** | 50 |
| **No. of Contact Hours/week** | 3 | **Total** | 100 |
| **Credits** | 3 | **Exam. Duration** | 3 Hours |

**Course objective is to:**

The purpose of this course is to make students well conversant with numerical methods to solve ordinary differential equations, sampling theory and Operational research emerging in science and engineering.

| **Module-1** | **L1,L2, L3** | **12 Hours** |
|---|---|---|

Numerical Methods-1
Numerical solution of Ordinary Differential Equations of first order and first degree: Modified Euler's method, Taylor's series method, Runge-Kutta method of fourth order, Predictor and Corrector method: Milne's Method and Adams-Bashforth Method.

Application: Solving Ordinary Differential Equations.

Video Links:
1. http://nptel.ac.in/courses.php?disciplineID=111
2. http://www.class-central.com/subject/math(MOOCs)
3. http://academicearth.org/

| **Module-2** | **L1,L2, L3** | **12 Hours** |
|---|---|---|

Numerical Methods-2:
Numerical solution of Ordinary Differential Equations of second order: Runge-Kutta method of fourth order, Predictor and Corrector method: Milne's Method and Adams Bashforth Method.

Calculus of Variations: Variation of function and Functional, variational problems.
Euler's equation, Geodesics.

Application: Hanging chain problem.

Video Links:
1. http://nptel.ac.in/courses.php?disciplineID=111
2. http://www.class-central.com/subject/math(MOOCs)
3. http://academicearth.org/

| Module-3 | L1,L2, L3 | 12 Hours |
|---|---|---|
| Operations Research-1<br>Introduction to Linear Programming Problem (LPP): Prototype example, Assumptions of LPP, Formulation of LPP and Graphical method various examples. The simplex method, Big M method, Two phase method and dual simplex method.<br><br>Application: Graphical solution procedure.<br><br>Video Links:<br>1. http://nptel.ac.in/courses.php?disciplineID=111<br>2. http://www.class-central.com/subject/math(MOOCs)<br>3. http://academicearth.org/ | | |
| **Module-4** | **L1,L2, L3** | **12 Hours** |
| Operations Research-2<br>The transportation problem: Initial Basic Feasible Solution (IBFS) by North West Corner Rule method, Matrix Minima Method, Vogel's Approximation Method.<br>Game Theory: The formulation of two persons, zero sum games; saddle point, maxmin and minmax principle, Solving simple games- a prototype example, Games with mixed strategies.<br><br>Application: Transportation problem.<br><br>Video Links:<br>1. http://nptel.ac.in/courses.php?disciplineID=111<br>2. http://www.class-central.com/subject/math(MOOCs)<br>3. http://academicearth.org/ | | |
| **Module-5** | **L1,L2, L3** | **12 Hours** |
| Statistical Methods<br>Correlation and Regression: Correlation, Regression coefficients, line of regression problems.<br><br>Curve fitting: Fitting of the curves of the form $y = ax + b$, $y = ax^2 + bx + c$, $y = ae^{bx}$ by the method of least squares.<br><br>Application: Finding the best fit between two variables.<br><br>Video Links:<br><br>1. http://nptel.ac.in/courses.php?disciplineID=111<br>2. http://www.class-central.com/subject/math(MOOCs)<br>3. http://academicearth.org/ | | |

| **Course outcomes:** | |
|---|---|
| CO1 | Solve first and second order ordinary differential equation arising in flow problems using single step numerical methods. |
| CO2 | Determine the extremals of functional and solve the simple problems of the Calculus of variations. |
| CO3 | Solve the mathematical formulation of linear programming problem. |
| CO4 | Solve the applications of transport problems and theory of games. |

| | |
|---|---|
| CO5 | Fit a suitable curve by the method of least squares and determine the lines of regression for a set of statistical data. |

| **Reference Books:** | |
|---|---|
| 1. | B.S. Grewal, "Higher Engineering Mathematics" Khanna Publishers, 43rd Edition, 2013. |
| 2. | S. D. Sharma, "Operations Research", Kedar Nath and Ram Nath Publishers, Seventh Revised Edition 2014. |
| 3. | Erwin Kreyszig, "Advanced Engineering Mathematics", Wiley-India publishers, 10th edition, 2014. |
| 4. | Ramana B. V., "Higher Engineering Mathematics", Tata Mc Graw-Hill, 2006. |
| 5. | Bali N. P. & Manish Goyal, "A text book of Engineering Mathematics", Laxmi Publications, 8th Edition |
| 6. | Erwin Kreyszig, "Advanced Engineering Mathematics", Wiley-India publishers, 10th edition, 2014. |

| **CO-PO Mapping** | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
| CO1 | 3 | 3 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| CO2 | 3 | 2 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CO3 | 3 | 3 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CO4 | 2 | 3 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| CO5 | 3 | 3 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

High-3, Medium-2, Low-1

| | | | |
|---|---|---|---|
| **Course Title** | Design & Analysis of Algorithm | **Semester** | IV |
| **Course Code** | MVJ19IS42 | **CIE** | 50 |
| **Total No. of Contact Hours** | 60 L : T : P :: 40 : 0 : 20 | **SEE** | 50 |
| **No. of Contact Hours/week** | 4 | **Total** | 100 |
| **Credits** | 4 | **Exam. Duration** | 3 Hours |

**Course objective is to:**

- Explain various computational problem-solving techniques.
- Apply appropriate method to solve a given problem.
- Describe various methods of algorithm analysis

| **Module-1** | **L1,L2, L3** | **12 Hours** |
|---|---|---|

Introduction to Algorithms: The role of algorithms in computing, Growth of functions, Asymptotic notations, Designing and Analysing algorithms-an Introduction using insertion sort. Review on the Math needed for algorithm design and analysis.

Laboratory Sessions/ Experimental learning:

- Implement insertion sort and test its efficiency

Applications: Develop a realistic model for the input to the program. Analyse the unknown quantities, assuming the modelled input. Calculate the total running time by multiplying the time by the frequency for each operation, then adding all the products.

Video link / Additional online information :
https://www.tutorialspoint.com/data_structures_algorithms/asymptotic_analysis.htm

| **Module-2** | **L1,L2, L3** | **12 Hours** |
|---|---|---|

Divide and Conquer: Solving recurrences – The Substitution method, Recurrence Tree method and Master's method, Multiplying large integers, Binary Search, Sorting [Merge Sort and Quick Sort], Selection in linear time [Expected and Worst-case], Strassen's algorithm for Matrix Multiplication, The maximum sub-array problem.

Laboratory Sessions/ Experimental learning:

- Implement maximum sub array algorithm and test their correctness and efficiency

Applications: Closest Pair of Points, Strassen's Multiplication, Karatsuba Algorithm, Cooley-Tukey Algorithm

Video link / Additional online information :
https://www.tutorialspoint.com/design_and_analysis_of_algorithms/design_and_analysis_of_algorithms_divide_conquer_htm

| Module-3 | L1,L2, L3 | 12 Hours |
|---|---|---|

Greedy Algorithms: Characteristics of Greedy algorithms, The problem of making change, Greedy algorithms for Scheduling, Minimum Spanning Trees – Kruskal's Algorithm and Prim's Algorithm, Greedy Algorithms for finding the shortest paths in a Graph, The Knapsack problem Amortized Analysis: The accounting method, The potential method.

Laboratory Sessions/ Experimental learning:

- Implement Knapsack Algorithm using Greedy method.

Applications: Dijkstra's Algorithm, Google Map

Video link / Additional online information :
https://www.tutorialspoint.com/design_and_analysis_of_algorithms/design_and_analysis_of_algorithms_greedy_method_htm

| Module-4 | L1,L2, L3 | 12 Hours |
|---|---|---|

Dynamic Programming: Calculating the binomial co-efficient, the problem of making change, The Knapsack problem, Chained matrix multiplication, Finding the shortest paths in a Graph, Reformulating Dynamic programming algorithms using recursion and memory functions.

Laboratory Sessions/ Experimental learning:

- Implement single source shortest path algorithm.

Applications: Logistic/Transportation Problems

Video link / Additional online information :
https://www.tutorialspoint.com/design_and_analysis_of_algorithms/design_and_analysis_of_algorithms_dynamic_programming_htm

| Module-5 | L1,L2, L3 | 12 Hours |
|---|---|---|

Backtracking: N-Queen's Problem -Graph colouring.

Branch and Bound: Assignment Problem - Traveling Salesman Problem. Computability classes – P, NP, NP-complete and NP-hard.

Laboratory Sessions/ Experimental learning:

- Implement graph colouring Problem

Applications: Electrical Engineering, Robotics, Artificial Intelligence, Materials Engineering, Solving Puzzles

Video link / Additional online information :
https://www.tutorialspoint.com/design_and_analysis_of_algorithms/design_and_analysis_of_algorithms_p_np_class_htm

| **Course outcomes:** | |
|---|---|
| CO1 | Analyze the correctness of algorithms using induction and loop invariants. |
| CO2 | Construct algorithms using design paradigms like divide and conquer, greedy and dynamic programming for a given problem. |
| CO3 | Analyze how the performance of an algorithm is affected based on the choice of data structures the algorithm uses. |
| CO4 | Construct graph-based algorithms to solve engineering problems. |
| CO5 | Outline P and NP problems with the help of backtracking and branch and bound techniques |

| **Reference Books:** | |
|---|---|
| 1. | Introduction to the Design and Analysis of Algorithms, Anany Levitin:, 2rd Edition, 2009.Pearson. |
| 2. | Computer Algorithms/C++, Ellis Horowitz, Satraj Sahni and Rajasekaran, 2nd Edition, 2014, Universities Press |
| 3. | Charles E. Leiserson, Thomas H. Cormen, Ronald L. Rivest, Clifford Stein – Introduction to Algorithms, Third edition, PHI, 2010. |

| **CO-PO Mapping** | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
| CO1 | 3 | 3 | 3 | 2 | 3 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| CO2 | 3 | 3 | 3 | 2 | 3 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| CO3 | 3 | 3 | 2 | 2 | 3 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| CO4 | 3 | 3 | 2 | 2 | 3 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| CO5 | 3 | 3 | 3 | 2 | 3 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |

High-3, Medium-2, Low-1

| | | | |
|---|---|---|---|
| **Course Title** | Microcontroller & Embedded Systems | **Semester** | IV |
| **Course Code** | MVJ19IS43 | **CIE** | 50 |
| **Total No. of Contact Hours** | 60 L : T : P :: 40 : 0 : 20 | **SEE** | 50 |
| **No. of Contact Hours/week** | 3 | **Total** | 100 |
| **Credits** | 3 | **Exam. Duration** | 3 Hours |

**Course objective is to:**

- Explain the fundamentals of ARM based system, basic hardware components, selection methods and attributes of an ARM Controller.
- Program ARM controller using the various instructions.
- Explain the fundamentals of Exception, Interrupt Handling and Memory Management Unit of ARM Controller.
- Identify the Embedded System Design applications.
- Explain the real time operating system for the embedded system design.

| **Module-1** | **L1,L2, L3** | **12 Hours** |
|---|---|---|

Microprocessors versus Microcontrollers, ARM Embedded Systems: The RISC design philosophy, The ARM Design Philosophy, Embedded System Hardware, Embedded System Software.

ARM Processor Fundamentals: Registers, Current Program Status Register, Pipeline, Exceptions, Interrupts, and the Vector Table , Core Extensions

Laboratory Sessions/ Experimental learning:

- ARM Processor and Sample programs using Simulator.
- Comparison of Microprocessor and Microcontroller hardware Model
- Comparing the Microprocessor and Microcontroller Software Model

Applications: ARM Design

Video link / Additional online information :
https://developer.arm.com/architectures/platform-design/embedded-systems
https://www.youtube.com/watch?v=JPfG0UQd3x4
https://bnmbiw.wordpress.com/2013/01/27/chapter-1-arm-embedded-systems/

| **Module-2** | **L1,L2, L3** | **12 Hours** |
|---|---|---|

Introduction to the ARM Instruction Set : Data Processing Instructions , Programme Instructions, Software Interrupt Instructions, Program Status Register Instructions, Coprocessor Instructions, Loading Constants

ARM programming using Assembly language: Writing Assembly code, Profiling and cycle counting, instruction scheduling

Laboratory Sessions/ Experimental learning:

- ARM assembly language programming

Applications: Writing Assembly code

Video link / Additional online information :
https://iitd-plos.github.io/col718/ref/arm-instructionset.pdf
https://www.slideshare.net/MathivananNatarajan/arm-instruction-set-60665439
https://www.scribd.com/document/401460874/ARM-Architecture

| **Module-3** | **L1,L2, L3** | **12 Hours** |
|---|---|---|

Exception, Interrupt Handling : Exception handling,Interrupts,Interrupt handling Schemes

Memory Management Unit : The Memory Hierarchy and Cache Memory, Cache Architecture, Cache Policy, Moving from MPU to an MMU,How Virtual Memory Works, Details of ARM MMU

Laboratory Sessions/ Experimental learning:

- Use of External interrupt0 to turn ON/OFF led connected to Pin P1.25 of ARM Processor.
- Use of Software Interrupt SWI instruction in programming.
- Calculating physical memory address from logical address.

Applications: Estimation of CPU & Memory Performance

Video link / Additional online information :
https://www2.seas.gwu.edu/~bhagiweb/cs211/lectures/cache1.pdf
https://developer.arm.com/docs/den0024/a/the-memory-management-unit
https://www.youtube.com/watch?v=IyRNk5SMEpM

| **Module-4** | **L1,L2, L3** | **12 Hours** |
|---|---|---|

Embedded System Components: Embedded Vs General computing system, History of embedded systems, Classification of Embedded systems, Major applications areas of embedded systems, purpose of embedded systems

Core of an Embedded System including all types of processor/controller, Memory, Sensors, Actuators, LED, 7 segment LED display, stepper motor, Keyboard, Push button switch, Communication Interface (on board and external types), Embedded firmware, Other system components.

Laboratory Sessions/ Experimental learning:

- Mini project

Case Study: Digital Clock, Battery operated Smartcard Reader

Applications: Displaying digits on a 7-segment LED interface

Video link / Additional online information :
https://www.slideshare.net/MoeMoeMyint/introduction-to-embedded-system-chapter-2-4th-portion
https://shrishailbhat.com/2018/02/28/arm-microcontroller-embedded-systems-embedded-system-components/
https://mrcet.com/downloads/digital_notes/ECE/IV%20Year/EMBEDDED%20SYSTEMS%20DESIGN.pdf

| **Module-5** | **L1,L2, L3** | **12 Hours** |
|---|---|---|

Real Time Operating System (RTOS) based Embedded System Design: Operating System basics, Types of operating systems, Task, process and threads (Only POSIX Threads with an example program), Thread pre-emption, Multiprocessing and Multitasking, Task Communication (without any program), Task synchronization issues – Racing and Deadlock, Concept of Binary and counting semaphores (Mutex example without any program), How to choose an RTOS

Case Study: Automated Meter Reading System (AMR) and Digital Camera, Real time concepts

Applications: Modern electronic systems

Video link / Additional online information :

https://www.geeksforgeeks.org/mutex-lock-for-linux-thread-synchronization/
http://digitalthinkerhelp.com/real-time-operating-system-rtos-examples-applications-functions/

| **Course outcomes:** | |
|---|---|
| CO1 | Describe the architectural features and instructions of ARM microcontroller |
| CO2 | Develop Assembly Programs in ARM for Embedded applications. |
| CO3 | Describe the fundamentals of Exception, Interrupt Handling and Memory Management Unit of ARM Controller |
| CO4 | Interface external devices and I/O with ARM microcontroller. |
| CO5 | Demonstrate the need of real time operating system for embedded system applications |

| **Reference Books:** | |
|---|---|
| 1. | Andrew N Sloss, Dominic Symes and Chris Wright, ARM system developer's guide, Elsevier, Morgan Kaufman publishers, 2008. |
| 2. | Shibu K V, "Introduction to Embedded Systems", Tata McGraw Hill Education, Private Limited, 2nd Edition. |
| 3. | Raghunandan...G.H, Microcontroller (ARM) and Embedded System, Cengage learning Publication, 2019 |
| 4. | The Insider's Guide to the ARM7 Based Microcontrollers, Hitex Ltd., 1st edition, 2005. |
| 5 | Steve Furber, ARM System-on-Chip Architecture, Second Edition, Pearson, 2015. |

| | |
|---|---|
| 6 | Raj Kamal, Embedded System, Tata McGraw-Hill Publishers, 2nd Edition, 2008. |

| **CO-PO Mapping** | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
| CO1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CO2 | 3 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CO3 | 0 | 0 | 2 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CO4 | 0 | 0 | 2 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CO5 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

High-3, Medium-2, Low-1

| | | | |
|---|---|---|---|
| **Course Title** | Object Oriented Programming Concepts | **Semester** | IV |
| **Course Code** | MVJ19IS44 | **CIE** | 50 |
| **Total No. of Contact Hours** | 60 L : T : P :: 40 : 0 : 20 | **SEE** | 50 |
| **No. of Contact Hours/week** | 3 | **Total** | 100 |
| **Credits** | 3 | **Exam. Duration** | 3 Hours |

**Course objective is to:**

- Learn fundamental features of object-oriented language and JAVA
- Design, write, debug, run C++ and Java Programs
- Develop console -based applications using C++
- Develop console & windows applications using Java.
- Introduce event driven Graphical User Interface (GUI) programming using applets and swings

| **Module-1** | **L1,L2, L3** | **12 Hours** |
|---|---|---|

Overview of OOPs Principles , Introduction to classes & objects , Instantiating and Using Classes with objects , Data Members , Member Functions , this Pointer , Constructor & Destructor , Control, Structures , Arrays in C++ .

Laboratory Sessions/ Experimental learning:

- Introduction to OOP lab (Simple C program) - Classes and Objects.

Applications: Building a secure program using data hiding concept . Using same function or same operator having different purposes

Video link / Additional online information :
http://ee402.eeng.dcu.ie/introduction/chapter-1---introduction-to-object-oriented-programming
https://introprogramming.info/english-intro-csharp-book/read-online/chapter-20-object-oriented-programming-principles/

| **Module-2** | **L1,L2, L3** | **12 Hours** |
|---|---|---|

Derived Class and Base Class , Derived Class Constructors , Overriding Member Functions , Public and Private Inheritance , Types of Inheritance: Single, Multi-Level, Multiple, Hierarchical and Hybrid , Virtual Base Classes , Abstract Classes.

Laboratory Sessions/ Experimental learning:

- Programs using constructor, inheritance

Applications: Reuse of existing class to derive a new class such that the redundant code is eliminated, which saves time and cost of program.

Video link / Additional online information :
https://isocpp.org/wiki/faq/private-inheritance
https://www.programiz.com/cpp-programming/public-protected-private-inheritance
https://balututorial.com/inheritance-in-c-with-example-program/

| Module-3 | L1,L2, L3 | 12 Hours |
|---|---|---|

Pointers , this Pointer , Pointers to Objects and Derived Classes , Function Overloading , Operator Overloading , virtual function,
Friend Function , Static Function ,Streams: Stream Classes - Unformatted I/O Operations - Formatted Console I/O Operation.

Laboratory Sessions/ Experimental learning:

- Program using function overloading, friend function

Applications: Dynamic linkage or late binding on the function

Video link / Additional online information :
https://www.cet.edu.in/noticefiles/285_OOPS%20lecture%20notes%20Complete.pdf
https://www.programiz.com/cpp-programming/friend-function-class
https://www.ntu.edu.sg/home/ehchua/programming/cpp/cp6_Inheritance.html

| Module-4 | L1,L2, L3 | 12 Hours |
|---|---|---|

Java Basics , Classes and Objects , Inheritance, Interfaces , Abstract Class , packages , Exception handling, Type casting

Laboratory Sessions/ Experimental learning:

- Programs using Java class/object, Package, interface

Applications: Partial abstraction with abstract classes. Total abstraction with interfaces

Video link / Additional online information :
https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/
https://www.edureka.co/blog/object-oriented-programming/

| Module-5 | L1,L2, L3 | 12 Hours |
|---|---|---|

Garbage Collections , Java Utility Classes , I/O Classes and Interfaces, Multithreading, Java swing basics

Laboratory Sessions/ Experimental learning:

- Programs using thread concept, Java swing

Applications: Partitioning the work of a project based on thread/objects.

Video link / Additional online information :
https://www.studytonight.com/java/garbage-collection.php
https://beginnersbook.com/2013/05/java-interface/

| |
|---|
| https://www.javatpoint.com/java-swing |

| **Course outcomes:** | |
|---|---|
| CO1 | Design class and objects for real world scenario. |
| CO2 | Apply Inheritance concept to obtain code reusability. |
| CO3 | Create applications to manipulate data from files using functions and streams |
| CO4 | Develop console applications using Java OOPS. |
| CO5 | Develop GUI application using Java library classes. |

| **Reference Books:** | |
|---|---|
| 1. | E Balagurusamy, Object Oriented Programming with C++, Tata McGraw Hill Publishing, New Delhi, 2011 |
| 2. | Sourav Sahay, Object Oriented Programming with C++ , 2nd Ed, Oxford University Press,2006 |
| 3. | Robert Lafore, Object Oriented Programming in C++, Galgotia Publication, 2010. |
| 4. | Herbert Schildt, Java: The Complete Reference, Eleventh Edition, McGraw-Hill Education,2018 |
| 5. | D.T. Editorial Services ,Java 8 Programming Black Book , second edition, Dreamtech Press,2015 |

| **CO-PO Mapping** | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
| CO1 | 0 | 2 | 2 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CO2 | 0 | 2 | 2 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CO3 | 0 | 2 | 1 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CO4 | 0 | 1 | 2 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CO5 | 0 | 1 | 2 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

High-3, Medium-2, Low-1

| | | | |
|---|---|---|---|
| **Course Title** | Operating System | **Semester** | IV |
| **Course Code** | MVJ19IS45 | **CIE** | 50 |
| **Total No. of Contact Hours** | 60 L: T : P :: 40 : 0 : 20 | **SEE** | 50 |
| **No. of Contact Hours/week** | 3 | **Total** | 100 |
| **Credits** | 3 | **Exam. Duration** | 3 Hours |

**Course objective is to:**

- Provide an understanding on the various components of an Operating System
- The course focuses on fundamental problems and optimal solutions for resource management in operating systems such as process, disk and memory management
- The course will introduce design principles and trade-offs in the design of Operating Systems.
- Explain inter-process communication.
- The course will also introduce the interface for interacting with a contemporary Operating system such as Linux.

| **Module-1** | **L1,L2, L3** | **12 Hours** |
|---|---|---|

Introduction to operating systems, What operating systems do; Computer System organization; Computer System architecture; Operating System operations; Process management; Memory management; Storage management; Protection and Security; Operating System Services; User - Operating System interface; System calls; Types of system calls; System programs; Operating system design and implementation; Operating System structure; Virtual machines. Process Management Process concept; Process scheduling; Operations on processes.
Case study:-IPC System

Laboratory Sessions/ Experimental learning:

- Implementing process scheduling algorithms

Applications: Computer system.

Video link / Additional online information:
https://nptel.ac.in/courses/106108101/

| **Module-2** | **L1,L2, L3** | **12 Hours** |
|---|---|---|

Multi-threaded Programming: Overview; Multithreading models; Thread Libraries; Threading issues. Process Synchronization: The critical section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors. CPU Scheduling: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Thread scheduling; Multiple-processor scheduling

Laboratory Sessions/ Experimental learning:

- Implementing process scheduling algorithms

Applications: spell-check, response to keyboard, formatting
Video link / Additional online information :

https://www.smartzworld.com/notes/operating-systems-pdf-vtu-os/

| Module-3 | L1,L2, L3 | 12 Hours |
|---|---|---|

Deadlocks: System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock.
Memory Management: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation.
CASE STUDY: ARM architecture

Laboratory Sessions/ Experimental learning:

- Implement Bankers algorithm for Dead Lock Avoidance

Applications: Traffic gridlock

Video link / Additional online information :

https://www.smartzworld.com/notes/operating-systems-pdf-vtu-os/

| Module-4 | L1,L2, L3 | 12 Hours |
|---|---|---|

Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing. File System, File concept; Access methods; Directory and disk structure; File system mounting; File sharing; Protection;
Case study's: NFS and WAFL File system

Laboratory Sessions/ Experimental learning:

- Implement all page replacement algorithms

Applications: scientific applications

Video link / Additional online information :

https://www.smartzworld.com/notes/operating-systems-pdf-vtu-os/

| Module-5 | L1,L2, L3 | 12 Hours |
|---|---|---|

Secondary Storage Structures, Protection: Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management; Swap space management. Protection: Goals of protection, Principles of protection, Domain of protection, Access matrix, Implementation of access matrix, Access control, Revocation of access rights, Capability- Based systems.
Linux overview – Kernel Architecture – Process, memory, file and I/O management – Inter Process communication and synchronization – Security. Case study of UNIX.

Laboratory Sessions/ Experimental learning:

- Implementing disk scheduling algorithm

Applications: NAS, Hard disk

Video link / Additional online information :

https://nptel.ac.in/courses/106108101/

| Course outcomes: | |
|---|---|
| CO1 | Recognize the important computer system resources and the role of operating system in their management policies and algorithms. |
| CO2 | Understand various scheduling algorithms. |
| CO3 | Familiar with principles of deadlock and its prevention. To understand the concepts of file system interface. |
| CO4 | Identify use and valuate the storage management policies with respect to different storage Management technologies |
| CO5 | Identify the need to create the special purpose operating system. |

| Reference Books: | |
|---|---|
| 1. | Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, Operating System Concepts,Ninth Edition, John Wiley & Sons (ASIA) Pvt. Ltd, 2005. |
| 2. | A.S.Tanenbaum, Operating System : Design and Implementation,Prentice Hall of India, 1989. |
| 3. | J.L.Galvin and A.Silberschatz, Operating System Concepts,Addison-Wesley, 1998 |

| CO-PO Mapping | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
| CO1 | 0 | 2 | 2 | 2 | 0 | 0 | 2 | 0 | 2 | 2 | 0 | 3 |
| CO2 | 0 | 3 | 2 | 2 | 0 | 0 | 2 | 0 | 2 | 2 | 0 | 2 |
| CO3 | 0 | 3 | 2 | 2 | 0 | 0 | 2 | 0 | 2 | 2 | 0 | 2 |
| CO4 | 0 | 2 | 2 | 2 | 0 | 0 | 2 | 0 | 2 | 2 | 0 | 3 |
| CO5 | 0 | 3 | 2 | 2 | 0 | 0 | 2 | 0 | 2 | 2 | 0 | 2 |

High-3, Medium-2, Low-1

| | | | |
|---|---|---|---|
| **Course Title** | Theory of Computation | **Semester** | IV |
| **Course Code** | MVJ19IS46 | **CIE** | 50 |
| **Total No. of Contact Hours** | 60 L : T : P :: 40 : 0 : 20 | **SEE** | 50 |
| **No. of Contact Hours/week** | 3 | **Total** | 100 |
| **Credits** | 3 | **Exam. Duration** | 3 Hours |

**Course objective is to:**

- Acquire knowledge of Automata Theory as the basis of all computer science languages design
- Understand the concept of Context Free Grammars and Languages
- Learn the tools used for Lexical and Syntax analysis
- Acquire knowledge of optimization
- Enrich the knowledge in various phases of compiler ant its use

| **Module-1** | **L1,L2, L3** | **12 Hours** |
|---|---|---|

Introduction - Basic Mathematical Notation and techniques - Finite State systems - Basic Definitions - Finite Automaton - DFA & NDFA - Regular Languages- Regular Expression - Equivalence of NFA and DFA - Equivalence of NDFAs with and without input moves - Equivalence of finite Automaton and regular expressions - Minimization of DFA.

Laboratory Sessions/ Experimental learning:

- Problems on DFA/NFA, regular expression

Applications: Text processing, compilers, and hardware design.Recognizing the pattern using regular expressions.

Video link / Additional online information :
https://www.youtube.com/watch?v=NeI1oOFlVtU
https://www.youtube.com/watch?v=_Z3XdgpE6_4
https://www.geeksforgeeks.org/regular-languages-and-finite-automata-gq/

| **Module-2** | **L1,L2, L3** | **12 Hours** |
|---|---|---|

Grammar Introduction - Types of Grammar - Context Free Grammars and Languages - Derivations and Languages - Ambiguity - Relationship between derivation and derivation trees - Simplification of CFG - Elimination of Useless symbols - Unit productions - Null productions - Pushdown Automata – Definitions - Moves - Instantaneous descriptions - Deterministic pushdown automata - Equivalence of Pushdown automata.

Laboratory Sessions/ Experimental learning:

- Problems on CFG, pushdown automata

Applications: CFGs can be used in programming languages, to study human language and in Artificial Intelligence

.

Video link / Additional online information :
https://www.youtube.com/watch?v=ocLRMFr0TMI
https://www.geeksforgeeks.org/ambiguity-in-context-free-grammar-and-context-free-languages/
https://www.cis.upenn.edu/~jean/gbooks/tcbookpdf2.pdf

| **Module-3** | **L1,L2, L3** | **12 Hours** |
|---|---|---|

Introduction to Compiling - The grouping of phases - Compiler construction tools. The role of the lexical analyzer - Input buffering - Specification of tokens - Recognition of tokens - A language for specifying lexical analyzer

Laboratory Sessions/ Experimental learning:

- Problems on lexical analysis

Applications: Designing lexical analyzer of a compiler.

Video link / Additional online information :
http://www.vssut.ac.in/lecture_notes/lecture1422914957.pdf
https://rmd.ac.in/dept/cse/notes/6/CD/unit1.pdf

| **Module-4** | **L1,L2, L3** | **12 Hours** |
|---|---|---|

Syntax Analysis - The role of the parser - Context-free grammars - Writing a grammar - Top-down parsing - Bottom-up Parsing - SR parsers - LR parsers - Constructing an SLR (1) parsing table. Type checking - Type Systems - Specification of a simple type checker.

Laboratory Sessions/ Experimental learning:

- Problems on syntax analysis, Construction of parsing table

Applications: Designing the parsing phase of a compiler (Syntax Analysis).

Video link / Additional online information :
https://www.tutorialspoint.com/compiler_design/compiler_design_bottom_up_parser.htm
https://www.geeksforgeeks.org/bottom-up-or-shift-reduce-parsers-set-2/
https://www.includehelp.com/compiler-design/introduction-to-bottom-up-parser.aspx

| **Module-5** | **L1,L2, L3** | **12 Hours** |
|---|---|---|

Intermediate languages - Declarations - Assignment statements - Boolean expressions - Case statements -Backpatching - Procedure calls - Issues in the design of a code generator - The target machine - Run-time storage - management - Basic blocks and flow graphs - Next-use information - A simple code – generator - Register allocation and assignment - The dag representation of basic blocks - Generating code from DAGs

Laboratory Sessions/ Experimental learning:

- Problems on DAG representation

Applications: Generation of code for boolean expressions

Video link / Additional online information :
https://lecturenotes.in/notes/18736-note-for-compiler-design-cd-by-bineeth-kuriakose/13
http://www.vssut.ac.in/lecture_notes/lecture1422914957.pdf

| **Course outcomes:** | |
|---|---|
| CO1 | Construct finite automata for given pattern and find its equivalent regular expressions. |
| CO2 | Design and simplify context free grammar and find equivalent pushdown automata for given language. |
| CO3 | Generate the machine code considering the functionalities involved in different phases of the compilation process. |
| CO4 | Implement the parsing techniques including Bottom-up and Top-down parsing for the given programming construct described in Context Free Grammar |
| CO5 | Design code generators for the specified machine and apply the various optimization techniques to speed up the compilation time. |

| **Reference Books:** | |
|---|---|
| 1. | Hopcroft J E, Motwani R and Ullman J D, Introduction to Automata Theory, Languages and Computations, Second Edition, Pearson Education, 2012. |
| 2. | Alfred V Aho, Ravi Sethi Jeffrey D Ullman, Compilers- Principles, Techniques, and Tools, Third Edition, Pearson Education Asia, 2009. |
| 3 | Steven S Muchnick, Advanced Compiler Design and Implementation, Second Edition, Morgan Kaufmann Pulishers, 2008. |
| 4 | Raghavan V, Principles of Compiler Design, Third Edition, Tata Mc-Graw Hill Education Pvt. Ltd., New Delhi, 2009 |

| **CO-PO Mapping** | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
| CO1 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| CO2 | 2 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CO3 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| CO4 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| CO5 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |

High-3, Medium-2, Low-1

| | | | |
|---|---|---|---|
| **Course Title** | Design & Analysis of Algorithm Lab | **Semester** | IV |
| **Course Code** | MVJ19ISL47 | **CIE** | 50 |
| **Total No. of Contact Hours** | 40 L: T: P: 0: 0: 40 | **SEE** | 50 |
| **No. of Contact Hours/week** | 4 | **Total** | 100 |
| **Credits** | 2 | **Exam. Duration** | 3 Hours |

**Course objective is to:**

- Understanding the basic algorithm techniques
- Solve different algorithmic technique problems
- Synthesize the efficiency of the algorithms in common engineering design situation

| Sl No | Experiment Name | RBT Level | Hours |
|---|---|---|---|
| 1 | Implementation of Binary Search Trees | L3 | 4 |
| 2 | Implementation of merge and quick sort algorithms and test their correctness and efficiency | L3 | 4 |
| 3 | Implementation of Floyd-Warshall Algorithm and test their efficiency | L3 | 4 |
| 4 | Implementation of 0/1 Knapsack problem using<br>(a) Dynamic Programming method<br>(b)Greedy method. | L3 | 4 |
| 5 | (a) Implementation of all-Pairs Shortest Paths problem<br>(b) Implementation of Travelling Sales Person problem | L3 | 4 |
| 6 | Implementation and analysis of running time of eight-queen problem | L3 | 4 |
| 7 | Implementation of insertion and topological sorting and test their efficiency. | L3 | 4 |
| 8 | Program to find a subset of a given set S = {Sl, S2,.....,Sn} of *n* positive integers | L3 | 4 |
| 9 | Program to find all Hamiltonian Cycles in a connected undirected Graph | L3 | 4 |
| 10 | Mini Project /Case Presentation | L3 | 4 |

**Course outcomes:**

| | |
|---|---|
| CO1 | Analyze the complexities of various problems |
| CO2 | Apply different algorithmic design paradigms and methods of analysis |
| CO3 | Analyzing the different complexity for different algorithmic techniques |

| | |
|---|---|
| CO4 | Implement various algorithms in a high-level language |
| CO5 | Compare the performance of different algorithms for same problem |

---

| **CO-PO Mapping** | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
| CO1 | 3 | 3 | 3 | 2 | 3 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| CO2 | 3 | 3 | 3 | 2 | 3 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| CO3 | 3 | 3 | 2 | 2 | 3 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| CO4 | 3 | 3 | 2 | 2 | 3 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| CO5 | 3 | 3 | 3 | 2 | 3 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |

High-3, Medium-2, Low-1

| | | | |
|---|---|---|---|
| **Course Title** | Microcontroller & Embedded Systems Lab | **Semester** | IV |
| **Course Code** | MVJ19ISL48 | **CIE** | 50 |
| **Total No. of Contact Hours** | 40 L: T: P: 0: 0: 40 | **SEE** | 50 |
| **No. of Contact Hours/week** | 4 | **Total** | 100 |
| **Credits** | 2 | **Exam. Duration** | 3 Hours |

**Course objective is to:**

- Develop Assembly language programs for any real time scenario using Arm Microcontroller
- Demonstrate various real time application using ARM Microcontroller hardware.

| Sl No | Experiment Name | RBT Level | Hours |
|---|---|---|---|
| 1 | Write a program to find the sum of first 10 integer numbers. | L3 | 2 |
| 2 | Write a program to find factorial of a number. | 3 | 3 |
| 3 | Write a program to add an array of 16 bit numbers and store the 32 bit result in internal RAM | 3 | 3 |
| 4 | Write a program to find the square of a number (1 to 10) using look-up table. | 3 | 3 |
| 5 | Write a program to find the largest/smallest number in an array of 32 numbers | 3 | 3 |
| 6 | Write a program to arrange a series of 32 bit numbers in ascending/descending order | 3 | 3 |
| 7 | Write a program to count the number of ones and zeros in two consecutive memory locations | 3 | 3 |
| 8 | Write an ARM assembly program that checks if a 32-bit number is a palindrome. Assume that the input is available in r 3. The program should set r 4 to 1 if it is a palindrome, otherwise r 4 should have 0. A palindrome is a number which is the same when read from both sides. For example, 1001 is a 4 bit palindrome. | 3 | 3 |
| 9 | Display “Hello World” message using Internal UART | 3 | 3 |
| 10 | Interface and Control a DC Motor | 3 | 3 |
| 11 | Interface a Stepper motor and rotate it in clockwise and anti-clockwise direction | 3 | 3 |
| 12 | Interface a DAC and generate Triangular and Square waveforms. | 3 | 3 |
| 13 | Display the Hex digits 0 to F on a 7-segment LED interface, with an appropriate delay in Between | 3 | 3 |
| **STUDY EXPERIMENT**<br>Interface a 4x4 keyboard and display the key code on an LCD | | L3 | 2 |
| | | | |

| Course outcomes: | |
|---|---|
| CO1 | Describe the internal architecture of microcontroller systems, including counters, timers, ports, and memory |
| CO2 | Develop programs using ARM7TDMI/LPC2148. |
| CO3 | Test programs using ARM7TDMI/LPC2148 |
| CO4 | Conduct experiments on an ARM7TDMI/LPC2148 evaluation board using evaluation version of Embedded 'C' & Keil Uvision-4 tool/compiler. |
| CO5 | Interface a microcontroller system to user controls and other electronic systems. |

**CO-PO Mapping**

| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CO2 | 3 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CO3 | 0 | 0 | 2 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CO4 | 0 | 0 | 2 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CO5 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

High-3, Medium-2, Low-1