SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
ITUDLT1	0	0	3	3 HOURS	25	25	1.5

PYTHON FOR DATA SCIENCE LAB

Course Objectives:

This course is designed to enable the students to:

1. To be able to introduce core programming basics and program design with functions using Python programming language.
2. To understand a range of Object-Oriented Programming, as well as in-depth data and information processing techniques.
3. To understand the high-performance programs designed to strengthen the practical expertise.

S.No.	Experiments
1	Write a program to demonstrate different number data types in Python.
2	Write a program to perform different Arithmetic Operations on numbers in Python.
3	Write a program to create, concatenate and print a string and accessing sub-string from a given string.
4	Write a python script to print the current date in the following format “Sun May 29 02:26:23 IST 2017”.
5	Write a program to create, append, and remove lists in python.
6	Write a program to demonstrate working with tuples in python.
7	Write a program to demonstrate working with dictionaries in python.
8	Write a python program to find largest of three numbers.
9	Write a Python program to convert temperatures to and from Celsius, Fahrenheit. [Formula : c/5 = f-32/9]
10	Write a Python script that prints prime numbers less than 20.
11	Write a python program to find factorial of a number using Recursion.
12	Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is a right triangle (Recall from the Pythagorean Theorem that in a right triangle, the square of one side equals the sum of the squares of the other two sides).
13	Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.
14	Write a python program to define a module to find Fibonacci Numbers and import the module to another program.
15	Write a python program to define a module and import a specific function in that module to another program.
16	Write a Python class to convert an integer to a roman numeral.
17	Write a Python class to implement pow(x, n).
18	Write a Python class to reverse a string word by word.

TEXT BOOKS:

1. Core Python Programming, Wesley J. Chun, Second Edition, Pearson.

REFERENCES BOOKS:

1. Think Python, Allen Downey, Green Tea Press.
2. Introduction to Python, Kenneth A. Lambert, Cengage.
3. Python Programming: A Modern Approach, Vamsi Kurama, Pearson.

4. Learning Python, Mark Lutz, O'Really.

Course Outcomes:

At the end of this course the student can answer how to:

1. Student should be able to understand the basic concepts scripting and the contributions of scripting language.
2. Ability to explore python especially the object-oriented concepts, and the built in objects of Python.
3. Ability to create practical and contemporary applications such as TCP/IP network programming, Web applications, discrete event simulations.

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
IT205PPC02	0	0	4	4 HOURS	30	20	2

Python Programming Lab

Course Objectives:

This course is designed to enable the students to:

1. To be able to introduce core programming basics and program design with functions using Python programming language.
2. To understand a range of Object-Oriented Programming, as well as in-depth data and information processing techniques.
3. To understand the high-performance programs designed to strengthen the practical expertise.

S.No.	Experiments
1	Write a program to demonstrate different number data types in Python.
2	Write a program to perform different Arithmetic Operations on numbers in Python.
3	Write a program to create, concatenate and print a string and accessing sub-string from a given string.
4	Write a python script to print the current date in the following format "Sun May 29 02:26:23 IST 2017".
5	Write a program to create, append, and remove lists in python.
6	Write a program to demonstrate working with tuples in python.
7	Write a program to demonstrate working with dictionaries in python.
8	Write a python program to find largest of three numbers.
9	Write a Python program to convert temperatures to and from Celsius, Fahrenheit. [Formula : c/5 = f-32/9]
10	Write a Python script that prints prime numbers less than 20.
11	Write a python program to find factorial of a number using Recursion.
12	Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is a right triangle (Recall from the Pythagorean Theorem that in a right triangle, the square of one side equals the sum of the squares of the other two sides).
13	Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.
14	Write a python program to define a module to find Fibonacci Numbers and import the module to another program.
15	Write a python program to define a module and import a specific function in that module to another program.
16	Write a Python class to convert an integer to a roman numeral.
17	Write a Python class to implement pow(x, n).
18	Write a Python class to reverse a string word by word.

TEXT BOOKS:

1. Core Python Programming, Wesley J. Chun, Second Edition, Pearson.

REFERENCES BOOKS:

1. Think Python, Allen Downey, Green Tea Press.
2. Introduction to Python, Kenneth A. Lambert, Cengage.
3. Python Programming: A Modern Approach, Vamsi Kurama, Pearson.

4. Learning Python, Mark Lutz, O'Really.

Course Outcomes:

At the end of this course the student can answer how to:

1. Student should be able to understand the basic concepts scripting and the contributions of scripting language.
2. Ability to explore python especially the object-oriented concepts, and the built in objects of Python.
3. Ability to create practical and contemporary applications such as TCP/IP network programming, Web applications, discrete event simulations.

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
IT206PPE23	0	0	4	4 HOURS	30	20	2

Queuing Theory & Modeling Lab

Course Objectives:

This course is designed to enable the students to:

1. To provide necessary basic concepts in probability and random processes for applications such as random signals, linear systems in communication engineering.
2. To understand the basic concepts of probability, one and two dimensional random variables and to introduce some standard distributions applicable to engineering which can describe real life phenomenon.
3. To understand the basic concepts of random processes which are widely used in IT fields.
4. To understand the concept of queueing models and apply in engineering.
5. To understand the significance of advanced queueing models.
6. To provide the required mathematical support in real life problems and develop probabilistic models which can be used in several areas of science and engineering.

S.No.	Experiments
1	Infinite Capacity Markovian Models.
2	Finite Capacity Markovian Models.
3	Retrial Markovian Models for Bulk Arrival/Service.
4	Infinite Capacity Non Markovian Models.
5	Finite Capacity Non Markovian Models.
6	Non Markovian Models for Bulk Arrival/Service.
7	Discrete time Experiments.
8	Open Tandem Queue.
9	Tandem Queue with feedback.
10	Open Queueing Network Experiment.

TEXT BOOKS:

1. Ibe. O.C., “Fundamentals of applied Probability and Random Processes”, Elsevier, 1st Indian Reprint, 2007
2. Gross. D. and Harris. C.M., “Fundamentals of Queueing Theory”, Wiley Student edition,2004.

REFERENCES:

3. Robertazzi, “Computer Networks and Systems: Queueing Theory and performance evaluation”, Springer, 3RdEdition, 2006.
4. Taha. H.A., “Operations Research”, Pearson Education, Asia, 8th Edition, 2007.
5. Trivedi.K.S.,“Probability and Statistics with Reliability, Queueing and Computer Science applications”, John Wiley and Sons, 2nd Edition, 2002.

Course Outcomes:

At the end of this course the student can answer how to:

1. Understand the fundamental knowledge of the concepts of probability and have knowledge of standard distributions which can describe real life phenomenon.
2. Understand the basic concepts of one and two dimensional random variables and apply in engineering applications.
3. Apply the concept of random processes in engineering disciplines.
4. Acquire skills in analyzing queueing models.
5. Understand and characterize phenomenon which evolve with respect to time in a probabilistic manner.

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
IT206PPC01	0	0	4	4 HOURS	30	20	2

Computer Networks Lab

Course Objectives:

This course is designed to enable the students to:

1. Understand basic network models and Different transmission media used for data communication.
2. Understand the data link design issues and various data link protocols used for data transmission.
3. Comprehend different routing algorithms used for data transmission from source to destination in a network layer.
4. Know how internet addresses are configured and how internet protocols are used in connecting internet.

S.No.	Experiments
1	To configure the IP address for a computer connected to LAN and to configure network parameters of a web browser for the same computer.
2	a. Installing of internal modem and connecting to Internet. b. To configure WiFi for your PC.
3	Study of Stop and Wait Protocol.
4	Study of Go Back N Protocol.
5	Study of Selective Repeat Protocol.
6	Study of Networking Devices.
7	Study of LAN, MAN and WAN.
8	To practice the color code for different cables.
9	To construct Peer to Peer Topology.
10	To Construct Star Topology.

TEXT BOOKS

1. Data Communications and Networking – Behrouz A. Forouzan. TMH.
2. Computer Networks — Andrew S Tanenbaum, Pearson Education/PHI.
3. Data and Computer Communication by William Stalling (Pearson Education).

REFERENCE BOOKS

1. An Engineering Approach to Computer Networks-S.Keshav, 2nd Edition, Pearson Education.
2. Understanding communications and Networks, 3rd Edition, W.A. Shay, Thomson.
3. Computer Networking by Ed Tittel (Schaum’s series) (TMH).
4. Comer, “Computer Networks and Internets with Internet Applications”, Pearson Education.

Course Outcomes:

At the end of this course the student can answer how to:

1. By learning models students suggest appropriate network model for data communication.
2. Know how reliable data communication is achieved through data link layer.
3. Suggest appropriate routing algorithm for the network.
4. Provide internet connection to the system and its installation.

SUB CODE	L	T	P	DURATION/WEEK	IA	ESE	CREDITS
ITUCLT1	0	0	3	3 HOURS	25	25	1.5

DATA STURCTURE LAB

Course Objective

1. Understand and remember algorithms and its analysis procedure.
2. Introduce the concept of data structures through ADT including List, Stack, Queues.
3. To design and implement various data structure algorithms.
4. To develop application using data structure algorithms.
5. Compute the complexity of various algorithms.

List of Practical's

1. Write a C program for declaration, assignment, and accessing the arrays elements.
2. Write a C Program to Find Average Marks obtained by a class of 30 Students in a Test.
3. Write a C program to perform Array Insertion Operation.
4. Write a C program to perform Array Deletion Operation.
5. Write a C program to implement Linear Search.
6. Write a C program to implement Binary Search.
7. Write a C program to implement Bubble Sort.
8. Write a C program to implement Merging operation.
9. Write a program in C to create and display Singly Linked List.
10. Write a program in C to create a singly linked list of n nodes and count the number of nodes.
11. Write a program in C to insert a new node at the beginning of a Singly Linked List.
12. Write a program in C to insert a new node at the end of a Singly Linked List.
13. Write a program in C to insert a new node after a given location of Singly Linked List.
14. Write a program in C to delete first node of Singly Linked List.
15. Write a program in C to delete the last node of Singly Linked List.
16. Write a program in C to delete a node from the middle of Singly Linked List.
17. Write a program in C to search an existing element in a singly linked list.
18. Write C programs to implement the stack push operation using an array.
19. Write C programs to implement the stack pop operation using an array.

References books:

1. Lipschutz, "Data Structures with C" Schaum's Outline Series, TMH.
2. Horowitz and Sahani, "Fundamentals of data Structures", Galgotia Publication Pvt. Ltd.
3. R. Kruse etal, "Data Structures and Program Design in C", Pearson Education Asia.
4. A. M. Tenenbaum, "Data Structures using C & C++", Prentice-Hall of India Pvt. Ltd.
5. K Loudon, "Mastering Algorithms with C", Shroff Publisher & Distributors Pvt. Ltd.
6. Jean Paul Trembley and Paul G. Sorenson, "An Introduction to Data Structures with applications", McGraw Hill.
7. G A V Pai, "Data Structures and Algorithms", TMH.
8. G.S.Baluja, "Data Structures through C", Dhanpat Rai & Co.
9. Yashavant Kanetkar, "Data Structure Through C", BPB Publication.

Course Outcome

At the end of this lab session, the student will

1. Design programs using a variety of data structures such as Stacks, Queues, Array, Binary Trees, and Linked List.
2. Analyze and implement various kinds of searching and sorting techniques.

3. Have practical knowledge on the applications of data structures.
4. Design and analyze complexity of different algorithms.
5. Design advance data structure using non linear data structure.

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
IT205PPC01	0	0	4	4 HOURS	30	20	2

Database Management Systems Lab

Course Objectives:

This course is designed to enable the students to:

1. Introduce ER data model, database design and normalization.
2. Learn SQL basics for data definition and data manipulation.

S.No.	Experiments
1	Concept design with E-R Model.
2	Relational Model.
3	Normalization.
4	Practicing DDL commands.
5	Practicing DML commands.
6	Querying (using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.).
7	Queries using Aggregate functions, GROUP BY, HAVING and Creation and dropping of Views.
8	Triggers (Creation of insert trigger, delete trigger, update trigger).
9	Procedures.
10	Usage of Cursors.

TEXT BOOKS:

1. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata Mc Graw Hill, 3rd Edition
2. Database System Concepts, Silberschatz, Korth, McGraw Hill, V edition.

REFERENCES BOOKS:

1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, Elmasri Navrate, Pearson Education.
3. Introduction to Database Systems, C.J. Date, Pearson Education.
4. Oracle for Professionals, The X Team, S. Shah and V. Shah, SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL, Shah, PHI.
6. Fundamentals of Database Management Systems, M. L. Gillenson, Wiley Student Edition.

Course Outcomes:

At the end of this course the student can answer how to:

1. Design database schema for a given application and apply normalization.
2. Acquire skills in using SQL commands for data definition and data manipulation.
3. Develop solutions for database applications using procedures, cursors and triggers.

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
IT206PPE24	0	0	4	4 HOURS	30	20	2

Image Processing Lab

Course Objectives:

This course is designed to enable the students to:

1. To study the image fundamentals and mathematical transforms necessary for image processing.
2. To study the image enhancement techniques.
3. To study image restoration procedures.
4. To study the image compression procedures.

S.No.	Experiments
1	Display of Gray scale Images
2	Histogram Equalization.
3	Design of Non-linear Filtering.
4	Determination of Edge detection using Operators.
5	Filtering in frequency domain.
6	Display of color images.
7	2-D DFT and DCT.
8	Conversion between color spaces.
9	DWT of images.
10	Segmentation using watershed transform.

Text/Reference Books:

1. R.C. Gonzalez and R.E. Woods, Digital Image Processing, Second Edition, Pearson Education 3rd edition 2008.

2. Anil Kumar Jain, Fundamentals of Digital Image Processing, Prentice Hall of India.2nd edition 2004.

3. Murat Tekalp , Digital Video Processing" Prentice Hall, 2nd edition 2015.

Course Outcomes:

At the end of this course the student can answer how to:

1. Study the image fundamentals, mathematical transforms necessary for image processing.
2. About the various techniques of image enhancement, reconstruction, compression and segmentation.
3. Know sampling and reconstruction procedures.
4. Design image processing systems.

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
IT206PPE21	0	0	4	4 HOURS	30	20	2

Microprocessor & Microcontroller Lab

Course Objectives:

Students will be able to.

1. To expose students to the operation of typical microprocessor (8086) trainer kit.
2. To prepare the students to be able to solve different problems by developing different programs in assembly language.
3. To develop the quality of assessing and analyzing the obtained data.
4. To understand and hands on training of Interfacing of devices.

S.No.	Experiments with 8086 Microprocessor
1	To perform addition & subtraction of two 8 – bit hexadecimal numbers.
2	To perform addition & subtraction 16 – bit hexadecimal numbers.
3	To perform addition & subtraction 32 – bit hexadecimal numbers.
4	To perform addition & subtraction of two 8 – bit decimal numbers and store the result in DX register.
5	To perform addition & subtraction of two decimal digits n and m using ASCII code store the result in ASCII format. Where n and m are decimal number with single decimal digits.
6	To perform addition & subtraction of two decimal digits n and m using ASCII code store the result in ASCII format in CX-BX register. Where n and m are decimal number with two decimal digits.
7	To perform multiplication of n and m. Where n and m are hexadecimal numbers.
8	To perform division of 16 – bit number with 8-bit number.
9	To perform multiplication of two 8-bit numbers using ASCII code store the result in ASCII form in DX. Register.
10	To perform division of two 8-bit numbers using ASCII code store the result in ASCII form in DX register.
11	To solve Arithmetic equation 3AX+5DX+BP and store the result in CX register.
12	To solve Arithmetic equation (P*Q)+(R*S).
13	To add only positive number from 100 data bytes.
14	To write a program to add series of 20 bytes.
15	To find positive & negative byte from 100 data bytes.
16	To find largest & smallest byte from block of data.

S.No.	Experiments with 8051 Microcontroller
1	Data transfer/exchange between specified memory locations.
2	Largest/smallest from a series.
3	Sorting (Ascending/Descending) of data
4	Addition / subtraction / multiplication / division of 8/16 bit data.
5	Sum of a series of 8 bit data.
6	Multiplication by shift and add method.
7	Square / cube / square root of 8 bit data.
8	Matrix addition.

9	LCM and HCF of two 8 bit numbers.
10	Code conversion – Hex to Decimal/ASCII to Decimal and vice versa.
Interfacing experiments using 8051 Trainer kit and interfacing modules (At least two Experiments are mandatory)	
11	Time delay generation and relay interface.
12	Display (LED/Seven segments/LCD) and keyboard interface
13	ADC interface..
14	DAC interface with wave form generation.
15	Stepper motor and DC motor interface.
16	Realization of Boolean expression through port.

Reference Books:

1. IBM PC Assembly Language and Programming, P. Abel, 5th Edition, PHI/Pearson Education.
2. Introduction To Assembly Language Programming, SivaramaP.Dandamudi, Springer Int. Edition,2003.
3. The 8088 and 8086 Microprocessors: Programming , Interfacing,Software,Hardware and Application,4th edition,W.A.Triebel,A.Singh,N.K.Srinath,Pearson Education.

Course Outcomes:

On completion of this lab course the students will be able to:

1. Understand and apply the fundamentals of assembly level programming of microprocessors and microcontroller.
2. Work with standard microprocessor real time interfaces including GPIO, serial ports, digital-to-analog converters and analog-to-digital converters.
3. Troubleshoot interactions between software and hardware.
4. Analyze abstract problems and apply a combination of hardware and software to address the problem.
5. Design &Develop the solutions of problems using 8051 microcontroller.

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
ITUCLT2	0	0	3	3 HOURS	25	25	1.5

OBJECT ORIENTED PROGRAMMING WITH C++ LAB

Course Objectives:

1. To understand and Practice Programming Construct: Variable, Operators, Control Structures, Loop, Functions, learn the concept of class and object and develop classes for simple applications with C++.
2. To learn how to implement Constructors, copy constructors and destructor functions.
3. To learn how to overload functions and operators in C++.
4. To learn how to design C++ classes for code reuse and perform inheritance.
5. To learn working with files and handle exceptions in program.

List of Experiments:-

1. Write a program to display message using cout statement.
2. Write a program to calculate average of five numbers given by user.
3. Write a program to calculate compound interest given P, R and T.
4. Write a program to calculate factorial of a given number.
5. Write a program to generate n numbers of fibonacii series. Value of n should be provided by user.
6. Write a function to calculate the power of a number raised to another number using function. Write appropriate main() function to read and display the result.
7. Write a function factorial to calculate the factorial of a number, write appropriate main function also.
8. Write a function swap to swap the value of two integer variables. Write appropriate main function for the program.
9. Write a function to perform sorting using bubble sort algorithm. Use arrays to store the list of numbers. Also write main() function to read contents and display output.
10. Write a program to perform overloading of area function.
11. Write a program with overloaded volume function. Use volume function to calculate the volume of a cube, cone, sphere etc.
12. Write a program to calculate simple interest. Use default argument for rate. Write main function to exhibit the use of default argument.
13. Write a program to show the use of return by reference.
14. Write a program with at least one function made as inline.
15. Create a structure data type with data items roll number, name, and total marks.Write main function to read data for two students and also display the stored data.
16. Create a class named 'Student' with a string variable 'name' and an integer variable 'roll no'. Assign the value of roll no as '2' and that of name as "John" by creating an object of the class Student.
17. Write a program to print the area of a rectangle by creating a class named 'Area' having two functions.First function named as ‘readData' takes the length and breadth of the rectangle as parameters and the second function named as ‘calculateArea' returns the area of the rectangle. Length and breadth of the rectangle are entered through keyboard.
18. Write a program that would print the information (name, year of joining, salary, address) of three employees by creating a class named 'Employee'. The output should be as follows:

```
Name    Year of joining              Address
Ramesh         1994             64-C New Delhi
Sam            2000              68-D Bilaspur
John           1999             26-B-Banglore
```

19. Define a class to represent a bank account. Include the following members:

    Data members:

    1. Name of the depositor.
    2. Account number.
    3. Type of account.
    4. Balance amount in the account.

Member functions:

1. To assign initial values.
2. To deposit an amount.
3. To withdraw an amount after checking the balance.
4. To display the name and balance.

Write a main program to test the program.

20. Define a class to represent a bank account (FOR 100 CUSTOMERS). Include the following members:
    Data members:
    1. Name of the depositor.
    2. Account number.
    3. Type of account.
    4. Balance amount in the account.

    Member functions:
    1. To assign initial values.
    2. To deposit an amount.
    3. To withdraw an amount after checking the balance.
    4. To display the name and balance.

    Write a main program to test the program.
21. Create two classes DM and DB which store the value of distances. DM stores distances in meters and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use friend function to carry out the addition operation. The object that stores result may be a DM object or DB object depending on the units in which result is required.
22. Create a class time to store time in hours and minutes. Write a program that can read values for the class objects and add one object with another object storing the result in third object. Use object as arguments to sum() function and object return type from sum() function to assign the sum to the third object.
23. Write a program to demonstrate the use of Static Keyword. Write a program to print the names of students by creating a Student class. If no name is passed while creating an object of the Student class, then the name should be "Unknown", otherwise the name should be equal to the String value passed while creating the object of the Student class. Use “new” operator to create an instance of “name” member of class.
24. Write a program to add two complex number using a friend function. Use appropriate constructor function to initialize the object.
25. Write a program to initialize an object with another using copy constructor.
26. Write a program to show the highest scorer in a test out of three students who appeared in a exam. Use this pointer to refer to objects.
27. Write a program to show the highest scorer in a test out of three students who appeared in a exam. Use this pointer to refer to objects. (USER INPUT)
28. Write a program to illustrate the creation and destruction of objects.
29. Write a program to illustrate pointer to member and pointer to object concepts of 00P.
30. Create two objects of a class with two integer type members. Compare the two operators have same member values. Using overloaded = operator for comparison.
31. Create a class FLOAT that contains one float data member. Overload all the four arithmetic operators so that they operate on the objects of FLOAT.
32. Write a program to overload << and >> operators to display time object.
33. Write a program to overload <=, >== and == operator to compare time objects.
34. Write a program to perform overloading of function call operator.
35. Write a program to perform overloading of subscripting operator.
36. Write a Program to Concatenate two Strings Using Operator Overloading(+).
37. Write a program to convert time type object to an Integer value and integer type value to object of time type. Use appropriate data type conversion functions to perform the required conversion.
38. Write a program for an inventory of product in store. Use appropriate data type conversion functions to perform the required conversion.

39. Define a class Student with data members as rollno and name. Derive a class Fees from student that has a data member fees and functions to submit fees and generate receipt.. Derive another class Result from Student and display the marks and grade obtained by the student.

Reference Books:

1. Object Oriented Programming with C++ by M. P. Bhave, S. A. Patekar, Pearson Education
2. Object Oriented Programming With C++ by E. Balaguruswamy.
3. Object Oriented Programming in turbo C++ by Robert Lafore.
4. Programming with C++ by D. Ravichandan.
5. Programming with C++ (SOS) by Hubbard.

Course Outcomes:-

1. Understand the C++ language features. Use the control structure and data types in C++. Write simple programs using classes and objects.
2. Understand the concepts of arrays, pointers, references and use of dynamic allocation operators. Write simple programs to implement Constructor & destructor concepts.
3. Understand the concept of Operator overloading and type conversion. Write simple programs using overloaded operators.
4. Understand the concepts of inheritance and virtual functions. Write simple programs to implement inheritance and virtual functions.
5. Understand file handling concepts, generic class and I/O exception handling. Write small programs to implement file handling concepts and exception handling.

SUB CODE	L	T	P	DURATION/WEEK	IA	ESE	CREDITS
ITUDLT2	0	0	3	3 HOURS	25	25	1.5

OPERATING SYSTEMS LAB

Course Objectives

1. To learn the fundamentals of Operating Systems.
2. To learn the mechanisms of OS to handle processes and threads and their communication.
3. To learn the mechanisms involved in memory management in contemporary OS.
4. To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols.
5. To know the components and management aspects of concurrency management.

List of Experiments

1. Simulate the following CPU scheduling algorithms-

    a) FCFS b) SJF c) Round Robin d) Priority.

2. Write a C program to simulate producer-consumer problem using Semaphores.

3. Write a C program to simulate the concept of Dining-philosophers problem.

4. Write a C program to simulate the following contiguous memory allocation Techniques-

    a) Worst fit b) Best fit c) First fit.

5. Simulate all page replacement algorithms a)

    a) FIFO b) LRU c) OPTIMAL

6. Simulate all File Organization Techniques

    a) Single level directory b) Two level directory

7: Simulate Bankers Algorithm for Dead Lock Avoidance and dead lock prevention.

8. Write a program to simulate disk scheduling algorithms.

    a) FCFS b) SCAN c) C-SCAN

Reference Books

1. Milenkovic M. , “Operating System concepts”, MGH
2. Tanenbaubm A. S. “Operating System design and implementation” , PHI
3. Silberschartz A.and Patterson J.I. , “ Operating system concepts”, Wisley.
4. Stilling William “ Operating System “, Maxwell McMillan International Edition 1992.
5. Dectel H.N. , “An introduction to operating system “, Addision Wisley.

Course Outcomes

1. Create processes and threads.
2. Develop algorithms for process scheduling for a given specification of CPU utilization, Throughput, Turnaround Time, Waiting Time, and Response Time.
3. Specification of memory organization develops the techniques for optimally allocating memory to processes by increasing memory utilization and for improving the access time.
4. Design and implement file management system.
5. For a given I/O devices and OS (specify) develop the I/O management functions in OS as part of a uniform device abstraction by performing operations for synchronization between CPU and I/O controllers.

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
IT206PPE22	0	0	4	4 HOURS	30	20	2

Web Technology & E-Commerce Lab

Course Objectives:

This course is designed to enable the students to:

1. Understand the web technologies to create adaptive web pages for web application.
2. Use CSS to implement a variety of presentation effects to the web application.
3. Know the concept and implementation of cookies as well as related privacy concerns.
4. Develop a sophisticated web application that employs the MVC architecture.

S.No.	Experiments
1	Design the following static web pages required for a Training and placement cell web site. 1) Home Page 2) Login Page 3) Registration page
2	Design the following static web pages required for a Training and placement cell web site. 4) Company Details Page 5) Alumni Details Page 6) Placement Staff Details Page
3	Design the following static web pages required for a Training and placement cell web site. 7) Student personal Info Page 8) Student Academic Info page 9) Semester Wise Percentage & their Aggregate page
4	Validate login page and registration page using regular expressions.
5	Apply different font styles, font families, font colors and other formatting styles to the above static web pages.
6	Install wamp server and tomcat server, access above developed static web pages using these servers.
7	Write a servlet/PHP to connect to the database, Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration.
8	Write a JSP/PHP to connect to the database, Insert the details of the student academic information with student academic info page.
9	Write a JSP which does the following job: Authenticate the user when he submits the login form using the user name and password from the database.
10	Write a JSP to insert the student’s semester wise percentages and calculate aggregate and insert into database.
11	Write a JSP to search the students according to their aggregate and produce sorted list or according to their Enroll number.

List of Books:

1. Minoli and Minoli, Web technology and Commerce, TMH.
2. Web Technology, Achyut Godbole, Atul Kahate, TMH.
3. Principles of Web Design, Sklar, Cengage.
4. Electronic Commerce, Schneider, cengage Learn.
5. The E-Business revolution, Daniel Amor, PHI.
6. E-Government, Satyanarayana, PHI.
7. E-Commerce, Greenstein, TMH.
8. Koisur David : Electronic Commerce, Microsoft.
9. Ravi Kalakota : Frontiers of Electronic commerce.

Course Outcomes:

At the end of this course the student can answer how to:

1. Integrate frontend and backend web technologies in distributed systems.
2. Facilitate interface between frontend and backend of a web application.
3. Debug, test and deploy web applications in different web servers.
4. Migrate the web applications to the other platforms like .Net.