IPS Academy, Institute of Engineering & Science

(A UGC Autonomous Institute, Affiliated to RGPV, Bhopal)

Scheme Based on AICTE Flexible Curriculum

Department of Computer Science & Engineering

Bachelor of Technology (B.Tech.)[Computer Science & Engineering (IoT)]

V Semester

S. No.	Course Code	Course type	Course Title	Maximum Marks Allotted					Total Marks	Contact Hours per week			Total Credits
				Theory			Practical						
				End Sem	Mid Sem. Exam.	Quiz/ Assignment	End Sem	Term work Lab Work &Sessional		L	T	P	
1.	IO08	PCC	Microprocessor & Microcontroller	60	25	15	–	–	**100**	2	1	–	3
2.	IO09	PCC	Operating System	60	25	15	–	–	**100**	2	1	–	3
3.	IO10	PCC	Theory of Computation	60	25	15	–	–	**100**	2	1	–	3
4.	IO11	PCC	Software Engineering & Project Management	60	25	15	–	–	**100**	3	–	–	3
5.	HS05	HSMC	Humanities and Social Sciences Open Courses - I	60	25	15	–	–	**100**	2	–	–	2
6.	CB01	IFC	Interdisciplinary Foundation Course-II	60	25	15	–	–	**100**	2	–	–	2
7.	IO08(P)	LC	Microprocessor & Microcontroller Lab	–	–	–	60	40	**100**	–	–	4	2
8.	IO09(P)	LC	Operating System Lab	–	–	–	60	40	**100**	–	–	2	1
9.	IO02(P)	SBC	Programming in Python Lab	–	–	–	60	40	**100**	–	–	4	2
10.	IO01	PROJ	Mini Project	–	–	–	60	40	**100**	–	–	4	2
11.	MLC03	MLC	Environmental Studies	100	–	–	–	–		**1**	–	–	0
12.		PROJ	Internship-I	To be completed during semester break. It’s Evaluation/Credit to be added in Sixth Semester.									
		Total		**360**	**150**	**90**	**240**	**160**	**1000**	**14**	**3**	**14**	**23**

Humanities and Social Sciences Open Courses-I

(A) English Language Proficiency

(B) German Language

(C) French Language

(D) Japanese Language

(E) Soft skill and Interpersonal Communication

-Interdisciplinary Foundation Course-II (IFC-CB01)

(Offered by CSITCS Branch)

Block Chain Technology

1 Hr Lecture	**1 Hr Tutorial**	**2Hr Practical**
1 Credit	1 Credit	1 Credit

IPS Academy, Institute of Engineering & Science

(A UGC Autonomous Institute, Affiliated to RGPV, Bhopal)

Scheme Based on AICTE Flexible Curriculum

Department of Computer Science & Engineering

Bachelor of Technology (B.Tech.)[Computer Science & Engineering (IoT)]

V Semester

Sr. No.	Course Type	Course Code	Course Title	Scheme			Credits
				L	T	P	
1	PCC	IO08	Microprocessor & Microcontroller	2	1	–	3
2	PCC	IO09	Operating System	2	1	–	3
3	PCC	IO10	Theory of Computation	2	1	–	3
4	PCC	IO11	Software Engineering & Project Management	3	–	–	3
5	HSMC	HS05	Humanities and Social Sciences Open Courses - I	2	–	–	2
6	IFC	CB01	Interdisciplinary Foundation Course-II	2	–	–	2
7	LC	IO08(P)	Microprocessor & Microcontroller Lab	–	–	4	2
8	LC	IO09(P)	Operating System Lab	–	–	2	1
9	SBC	IO02(P)	Programming in Python Lab	–	–	4	2
10	PROJ	IO01	Mini Project	–	–	4	2
11	MLC	MLC03	Environmental Studies	1	–	–	Audit*
12	PROJ	–	Internship-I	Credit to be added in Sixth Semester			
Total Academic Engagement and Credits				**14**	**3**	**14**	**23**
				31			

* Zero credit course

- **Humanities and Social Sciences Open Courses – I**
  - (A) English Language Proficiency
  - (B) German Language
  - (C) French Language
  - (D) Japanese Language
  - (E) Soft skill and Interpersonal Communication

- **Interdisciplinary Foundation Course-II (IFC-CB01)**

(Offered by CSITCS Branch)

- ➢ Block Chain Technology

IPS Academy, Institute of Engineering & Science

(A UGC Autonomous Institute, Affiliated to RGPV, Bhopal)

Scheme Based on AICTE Flexible Curriculum

Department of Computer Science & Engineering

Bachelor of Technology (B.Tech.) [Computer Science & Engineering (IoT)]

V-Semester

PCC-IO-08	**Microprocessor and Microcontroller**	**2L:1T:0P (3hrs.)**	**3 Credits**

Prerequisite: Fundamentals of Digital Electronics and programming.

Course Objective: The objective of this course is that students can learn fundamental concepts of embedded system design and robotics and advanced microcontrollers like PIC, AVR and ARM.

Course Contents: (40 hrs.)

Module 1: (08hrs.)
Microprocessor 8086 Architecture - BIU and EU, Registers, Pin Diagram, Memory Addressing, Clock Generator 8284, Buffers and Latches, Maximum and Minimum Modes.

Module 2: (08hrs.)
Addressing Modes, Instruction set of 8086, Assembly Language Programming, Assemblers, Procedures, Macros, Interrupts, 8086 Based Multiprocessor Systems - Coprocessors (8087 NDP), Closely and Loosely Coupled Multiprocessor Systems (8089 IOP).

Module 3: (08hrs.)
Interfacing Chips-8251A (USART), 8255A (Programmable Peripheral Interface), 8253/8254 (Programmable Interval Timer/Counter), 8257 (DMA Controller), 8259A (Programmable Interrupt Controller).

Module 4: (08hrs.)
Microcontrollers - Microcontroller 8051- Architecture, Pin Diagram, I/O Ports, Internal RAM and Registers, Interrupts, Addressing Modes, Memory Organization and External Addressing, Instruction Set, Assembly Language Programming, Real Time Applications of Microcontroller- Interfacing with LCD, ADC, DAC, Stepper Motor, Key Board and Sensors.

Module 5:(08hrs.)
Embedded Systems- Introduction, Classification, Processors architecture, Hardware Units, Software embedded into System, Applications and products of embedded systems, Structural units in processor, Memory devices, I/O devices, Buses, Interfacing of processor memory and I/O devices, Case study of an embedded system for a smart card.

Course Outcome:

Students should be able to:

1. Understand the fundamentals and history of embedded system design.
2. Define different types of Actuators used in Robotics and illustrate concepts about their working.
3. Classify types of Robots in different applications and define various concepts related to their
   movements.
4. Understand PIC &amp; AVR microcontroller architectures and programming in Robotics and
   embedded systems.
5. Design automated embedded systems by interfacing different Modules with advance controllers. Illustrate overview of ARM microcontroller architectures.

List of Text/Reference Books:

1. B. B. Brey: The Intel Microprocessors, Architecture, Programming and Interfacing, Pearson Education.
2. Liu Gibson: Microcomputer Systems: The 8086/8088 Family- Architecture, Programming and Design PHI.
3. D. V. Hall: Microprocessors and Interfacing, TMH.
4. Mazidi and Mazidi: The 8051 Microcontroller and Embedded Systems, Pearson Education.
5. Ayala Kenneth:- The 8051 microcontroller, Third Edition, Cengage Learning.
6. V. Deshmukh: Microcontroller (Theory and Application), TMH.
7. Raj Kamal: Embedded Systems- Architecture , Programming and Design, TMH, New Delhi.
8. V. Udayashankara and M. S. Mallikarjuna swamy: 8051 Microcontroller, TMH, New Delhi.

IPS Academy, Institute of Engineering & Science

(A UGC Autonomous Institute, Affiliated to RGPV, Bhopal)
Scheme Based on AICTE Flexible Curriculum

Department of Computer Science & Engineering

Bachelor of Technology (B.Tech.) [Computer Science & Engineering (IoT)]

V-Semester

PCC-IO-09	**Operating System**	**2L:1T:0P (3hrs.)**	**3 Credits**

Prerequisite: Computer Organization & Architecture.

Course Objective: The objective of this course is to focus on process, memory & file system.

Course Contents: (40 hrs.)

Module 1: (06hrs.)
Introduction to Operating Systems: Function, Evolution, Different Types, Desirable Characteristics and features of an O/S, Operating Systems Services: Types of Services, Different ways of providing these Services – Utility Programs, System Calls, Operating System Structure, and Spooling & Buffering.

Module 2: (12hrs.)
CPU Scheduling: Process Concept, Scheduling Concepts, Types of Schedulers, Scheduling Criteria, Process State Diagram, Scheduling Algorithms, Operation on Process, Algorithms Evaluation, System calls for Process Management; Multiple Processor Scheduling; Concept of Threads. Concurrent Processes: Real and Virtual Concurrency, Mutual Exclusion, Synchronization, Inter- Process Communication, Critical Section Problem, Solution to Critical Section Problem: Semaphores – Binary and Counting Semaphores, WAIT & SIGNAL Operations and their implementation. Deadlocks: Deadlock Problems, Characterization, Prevention, Avoidance, Recovery.

Module 3: (10hrs.)
Memory Management: Different Memory Management Techniques – Partitioning, Swapping, Segmentation, Paging, Paged Segmentation, Comparison of these techniques, Techniques for supporting the execution of large programs: Overlay, Dynamic Linking and Loading, Virtual Memory – Concept, Implementation by Demand Paging etc., Page replacement algorithms.

Module 4: (08hrs.)

File Systems: File Concept, User's and System Programmer's view of File System, Disk Organization, Tape Organization, Different Modules of a File System, Disk Space Allocation Methods – Contiguous, Linked and Indexed. Directory Structures, File Protection, System Calls for File Management, Disk Scheduling Algorithms.

Module 5:(04hrs.)

Introduction to Network, Distributed and Multiprocessor Operating Systems. Case Studies: Unix/Linux, WINDOWS and other Contemporary Operating Systems.

Course Outcome:

1. State the core concepts of operating system, evolution and types of operating system.
2. Illustrate various input output concepts, inter process communication and deadlock
3. Illustrate process scheduling and memory management techniques.
4. Describe the concept of file and disk management.
5. State the core concepts of network, distributed and multiprocessor operating system.

List of Text / Reference Books:

1. Avi Silber Schatz, Peter Galvin, Greg Gagne, "Operating System Concepts Essentials", Wiley Asia Student Edition, 10th Edition, 2018.
2. William Stallings, "Operating Systems: Internals and Design Principles", Prentice Hall of India, 5th Edition, 2005.
3. Charles Crowley, "Operating System: A Design-oriented Approach", Irwin Publishing, 1st Edition.
4. Gary J. Nutt, "Operating Systems: A Modern Perspective", Addison-Wesley, 2nd Edition.
5. Maurice Bach, "Design of the Unix Operating Systems", Prentice-Hall of India, 8th Edition.
6. Daniel P. Bovet, Marco Cesati, "Understanding the Linux Kernel", O' Reilly and Associates, 3rd Edition.
7. Andrew S. Tanenbaum, "Modern Operating Systems", Prentice Hall, 3 rdEdition,2007.
8. Bovet & Cesati, "Understanding the Linux Kernel", O' Reily, 3rd Edition.

IPS Academy, Institute of Engineering & Science

(A UGC Autonomous Institute, Affiliated to RGPV, Bhopal)

Scheme Based on AICTE Flexible Curriculum

Department of Computer Science & Engineering

Bachelor of Technology (B.Tech.) [Computer Science & Engineering (IoT)]

V-Semester

PCC-IO-10	**Theory of Computation**	**2L:1T:0P (3hrs.)**	**3 Credits**

Prerequisite: Discrete structure.

Course Objective: The main objective of this course is to understand fundamental of theory of Computation.

Course Contents: (40hrs.)

Module 1: (08hrs.)

Introduction of Automata Theory: Examples of automata machines, Finite Automata as a language acceptor and translator, Moore machines and mealy machines, composite machine, Conversion from Mealy to Moore and vice versa.

Module 2: (08hrs.)

Types of Finite Automata: Non-Deterministic Finite Automata (NDFA), Deterministic finite automata machines, conversion of NDFA to DFA, minimization of automata machines, regular expression, Arden's theorem. Meaning of union, intersection, concatenation and closure, 2-way DFA.

Module 3: (08hrs.)

Grammars: Types of grammar, context sensitive grammar, and context free grammar, regular grammar. Derivation trees, ambiguity in grammar, simplification of context free grammar, conversion of grammar to automata machine and vice versa, Chomsky hierarchy of grammar, killing null and unit productions. Chomsky normal form and Greibach normal form.

Module 4: (08hrs.)

Push down Automata: example of PDA, deterministic and non-deterministic PDA, conversion of PDA into context free grammar and vice versa, CFG equivalent to PDA, Petrinet model.

Module 5: (08hrs.)

Turing Machine: Techniques for construction. Universal Turing machine Multitap, multihead and multidimensional Turing machine, N-P complete problems. Decidability and Recursively Enumerable Languages, decidability, decidable languages, undecidable languages, Halting problem of Turing machine & the post correspondence problem.

Course Outcomes

1. Explain the basic concepts of switching and finite automata theory & languages.
2. Relate practical problems to languages, automata, computability and complexity.
3. Construct abstract models of computing, check their power to recognize the languages and analyze the grammar, its types, simplification and normal form.
4. Interpret rigorously formal mathematical methods to prove properties of languages, grammars and automata.
5. Develop an overview of how automata theory, languages and computation are applicable in engineering application.

List of Text/ Reference Books:

1. DanielI. A. Cohen, “Introduction to Computer Theory”, Wiley India,2^{nd} Edition,2003.
2. John E Hopcroft, Jeffrey D. Ullman and Rajeev Motwani, “Introduction to Automata Theory, Languages and Computation”, Pearson Education, 2nd Edition,2001.
3. K. L. P Mishra & N. Chandra sekaran, “Theory of Computer Science”, PHI Learning, 3rd Edition,2006.
4. Peter Linz, “Introduction to Automata Theory and Formal Languages”, Narosa Publishing. 3rdEdition,2007.
5. John C Martin, “Introduction to languages and the theory of computation”, TATA McGraw Hill, 3rd Edition 2013.
6. Harry R. Lewis and Christos H. Papadimitriou, “Elements of the Theory of Computation”, Pearson Education Asia,2nd edition, 1998.
7. Dexter C. Kozen, “Automata and Computability”, Undergraduate Texts in Computer Science, Springer,1st edition,2012.
8. Michael Sipser, “Introduction to the Theory of Computation”, PWS Publishing.,3rd edition, 2012.

V-Semester

PCC-IO-11	Software Engineering & Project Management	3L:0T:0P (3hrs.)	3 Credits

Pre-requisite: Basic computer skills.

Course Objective: The purpose of this subject is to cover the underlying concepts and techniques used in Software Engineering & Project Management.

Course Contents: (40 hrs.)

Module 1: (08hrs.)

The Software Product and Software Process

Software Product and Process Characteristics, Software Process Models: Linear Sequential Model, Prototyping Model, RAD Model, Evolutionary Process Models like Incremental Model, Spiral Model, Component Assembly Model, RUP and Agile processes. Software Process customization and improvement, CMM, Product and Process Metrics, Feasibility Analysis, Cost Estimation Model.

Module 2: (08hrs.)

Requirement Elicitation, Analysis, and Specification

Functional and Non-functional requirements, Requirement Sources and Elicitation Techniques, Use case Modeling, System and Software Requirement Specifications, Requirement Validation, Traceability.

Module 3: (08hrs.)

Software Design

The Software Design Process, Design Concepts and Principles, Software Modeling and UML, Architectural Design, Architectural Views and Styles, User Interface Design, Function- oriented Design, SA/SD Component Based Design and Design Metrics.

Module 4:(08hrs.)

Software Analysis and Testing

Software Static and Dynamic analysis, Code inspections, Software Testing, Fundamentals, Software Test Process, Testing Levels, Test Criteria, Test Case Design, Test Oracles, Test Techniques, Black-Box Testing, White-Box Unit Testing and Unit, Testing Frameworks, Integration Testing, System Testing and other Specialized, Testing, Test Plan, Test Metrics, Testing Tools.

Module 5: (08hrs.)

Software Maintenance & Software Project Measurement

Need and Types of Maintenance, Software Configuration Management (SCM), Software Change Management, Version Control, Change control and Reporting, Program Comprehension Techniques, Re-engineering, Reverse Engineering, Tool Support. Project Management Concepts, Project and Process Planning, Resources Allocations, Project Scheduling and Tracking, Risk Assessment and Mitigation, Software Quality Assurance (SQA). Project Plan, Project Metrics.

Course Outcomes:

1. Decompose the given project in various phases of a lifecycle.
2. Choose appropriate process model depending on the user requirements.
3. Perform various life cycle activities like Analysis, Design, Implementation, Testing & Maintenance.
4. Know various processes used in all the phases of the product.
5. Apply the knowledge, techniques, and skills in the development of a software product

List of Text / Reference Books:

1. Pankaj Jalote, “An Integrated Approach to Software Engineering”, Narosa Pub, 2005.
2. Rajib Mall, “Fundamentals of Software Engineering” Second Edition, PHI Learning, Fourth Edition, 2014.
3. P, S. Pressman “Software Engineering. A Practitioner's Approach” New edition, McGraw Hills,7th edition,2010.
4. Sommer ville, “Software Engineering” , Pearson Education, 9th Edition,2011.
5. Richard H. Thayer, “Software Engineering & Project Managements”, Wiley India
6. Waman S. Jawadekar, Software Engineering, TMH,2004.
7. Bob Hughes, M. Cotterell, Rajib Mall “Software Project Management”, McGraw Hill, Sixth Edition,2017
8. Schwalbe, Kathy “Information Technology Project Management” 8th Edition, 2016.
9. Kieron Conway “Software project Management from concept to development Black Book” Dream tech Press.
10. Deepak Jain, “Software Engineering principle and practices” Oxford UniversityPress,2008.
11. Bell Douglas “Software Engineering for students” ,Pearson Education.,4th Edition,2005.
12. Kelkar “Software Project Management,” PHI Learning,3rd edition 2012.

V-Semester

IFC Offered by "CSE (IOT and Cyber Security including Blockchain Technology)" branch

CB01	**Block Chain Technology**	**2L:0T:0P (2hrs.)**	**2 Credits**

Pre-requisite: NA.

Course Objective: To understand the concept of Blockchain and its platforms- Bitcoin, Ethereum, Hyperledger and Multichain. The course provides an overview of the structure and mechanism of Blockchain.

Course Contents: (25 hrs.)

Module 1: (05hrs.)

Blockchain: Introduction and crypto foundation: Elliptic curve Cryptography, ECDSA, encryption and decryption. Introduction to Blockchain Technology with its Applications, Blockchain Network, Hashing Algorithm, SHA-256, Immutable Ledger, Centralized and Distributed P2P Network.

Module 2: (05hrs.)

Blockchain Mining: Blockchain Mining, Byzantine General Problem, Consensus Protocol and its types- Proof of work (PoW) and proof-of-stake (PoS) algorithm.

Module 3: (05hrs.)

Cryptocurrency: Bitcoin, Bitcoin addresses, Bitcoin Ecosystem, Bitcoin's Monetary Policy, The Halving Problem, Block frequency, The Nonce, difficulty adjustment algorithm, mining pools, transactions, Ethereum, overview of differences between Ethereum and bitcoin, block format, mining algorithm, account management, contracts, Solidity language, decentralized application using Ethereum.

Module 4:(05hrs.)

Smart Contract : Introduction to Smart Contracts, Different Blockchains and Consensus mechanisms. Smart contacts (escrow, micropayments, and decentralized lotteries), payment channels.

Module 5: (05hrs.)

Application Areas of Blockchain: Blockchain and its application with IOT and Cybersecurity, Blockchain and Security R3, CORDA and Hyperledger System architecture, ledger format, chain code, transaction flow and ordering, private channels, membership service providers, case studies.

Course Outcomes: After Completing the course student should be able to::

1. Describe the basic concepts blockchain technology.
2. Understand several types of consensus protocols.
3. Illustrate the concepts of Bitcoin along with different types of cryptocurrencies.
4. Understand the working and importance of smart contracts.
5. Analyze the block chain applications in a structured manner.

List of Text / Reference Books:

1. Mastering Bitcoin: Unblocking Digital Cryptocurrencies, by Andreas Antonopoulos.
2. Mastering Ethereum, Antonopoulos, Andreas M. and Wood, O'Reilly Media, Inc.,2018
3. An Introduction to Bitcoin, V. Saravanan, Lecture Notes.
4. Bitcoin and Cryptocurrencies Technologies: A Comprehensive Introduction, Arvind Narayanan, Princeton University Press (July 19,2016) ISBN-10:0691171696.

IPS Academy, Institute of Engineering & Science

(A UGC Autonomous Institute, Affiliated to RGPV, Bhopal)

Scheme Based on AICTE Flexible Curriculum

Department of Computer Science & Engineering

Bachelor of Technology (B.Tech.) [Computer Science & Engineering (IoT)]

V-Semester

LC-IO-08	Microprocessor and Microcontroller Lab	0L:0T:4P (4hrs.)	2 Credits

Prerequisite: Fundamentals of Digital Electronics and programming.

Course Objective: The objective of this course is that students can learn fundamental concepts of embedded system design and robotics and advanced microcontrollers like PIC, AVR and ARM.

Module 1:(04hrs.)
Microprocessor 8086 Architecture - BIU and EU, Registers, Maximum and Minimum Modes.

Module 2: (04hrs.)
Addressing Modes, Instruction set of 8086, Assembly Language Programming, Assemblers, Procedures,

Module 3: (04hrs.)
Interfacing Chips-8255A (Programmable Peripheral Interface), 8253/8254 (Programmable Interval Timer/Counter), 8257 (DMA Controller).

Module 4: (04hrs.)
Microcontrollers - Microcontroller 8051- Architecture, Pin Diagram, Interfacing with LCD, ADC, DAC, Stepper Motor, Keyboard and Sensors.

Module 5: (06hrs.)
Embedded Systems-Introduction, Classification, Processor's architecture, Interfacing of Processor Memory and I/O Devices.

Course Outcome:

Students should be able to:

1. Understand the fundamentals and history of embedded system design.
2. Define different types of Actuators used in Robotics and illustrate concepts about their working.
3. Classify types of Robots in different applications and define various concepts related to their movements.
4. Understand PIC &amp; AVR microcontroller architectures and programming in Robotics and embedded systems.
5. Design automated embedded systems by interfacing different Modules with advance

controllers. Illustrate overview of ARM microcontroller architectures.

List of Text/Reference Books:

1. B. B. Brey: The Intel Microprocessors, Architecture, Programming and Interfacing, Pearson Education.
2. Liu Gibson: Microcomputer Systems: The 8086/8088 Family- Architecture, Programming and
   Design PHI
3. D. V. Hall: Microprocessors and Interfacing, TMH.
4. Mazidi and Mazidi: The 8051 Microcontroller and Embedded Systems, Pearson Education.
5. Ayala Kenneth:- The 8051 microcontroller, Third Edition, Cengage Learning.
6. V. Deshmukh: Microcontroller (Theory and Application), TMH.
7. Raj Kamal: Embedded Systems- Architecture, Programming and Design, TMH, New Delhi.
8. V. Udayashankara and M. S. Mallikarjuna swamy: 8051 Microcontroller, TMH, New Delhi.

List of Experiments:

1. Assembly Language Programs of Microprocessor 8086.
2. Assembly Language Programs of Microcontroller 8051.
3. Assembly Language Programs for Interfacing Chips.
4. Write a program in embedded C to read temperature from LM35 and display on LCD.
5. Write a program in embedded C to read data from keypad &amp; display on LCD.
6. Write a program in embedded C to control speed of motor.
7. Write a program in embedded C to control servomotor.
8. Write a program in embedded C to control IR sensor.
9. Code a sequence in Robotic software to pick and place an object by Robotic hand.
10. Code a sequence in Robotic software to control hands of a humanoid Robot.
11. Code a sequence in Robotic software to control legs of a humanoid Robot.
12. Code a sequence in Robotic software to make a Robot walk.
13. Code a sequence in Robotic software to make a Robot dance.

IPS Academy, Institute of Engineering & Science

(A UGC Autonomous Institute, Affiliated to RGPV, Bhopal)
Scheme Based on AICTE Flexible Curriculum

Department of Computer Science & Engineering

Bachelor of Technology (B.Tech.) [Computer Science & Engineering (IoT)]

V-Semester

LC-IO-09	**Operating Systems Lab**	**0L:0T:2P (2 hrs.)**	**1 Credit**

Prerequisite: Computer system organization.

Course objective: This Course provides a comprehensive introduction of Operating System, Process Management, Memory Management, File Management and I/O management.

Module 1 (06hrs.)
Introduction to Operating Systems: Function, Evolution, Different Types, Desirable Characteristics and features of an O/S, Operating Systems Services.

Module 2:(04hrs.)
CPU Scheduling: Process Concept, Scheduling Concepts, Types of Schedulers, Scheduling Criteria, Process State Diagram, Scheduling Algorithms, Operation on Process, Algorithms Evaluation, System calls for Process Management; Multiple Processor Scheduling; Concept of Threads. Concurrent Processes: Real and Virtual Concurrency, Mutual Exclusion, Synchronization, Inter- Process Communication, Critical Section Problem, Solution to Critical Section Problem.

Module 3:(04 Hrs.)
Memory Management: Different Memory Management Techniques-Partitioning, Swapping, Segmentation, Paging, Paged Segmentation, Comparison of techniques.

Module 4: (06hrs.)
File Systems: File Concept, User's and System Programmer's view of File System, Disk Organization, Tape Organization, Different Modules of a File System, Disk Space Allocation Methods – Contiguous, Linked and Indexed. Directory Structures, File Protection, System Calls for File Management, Disk Scheduling Algorithms.

Module 5: (04hrs.)
Introduction to Network, Distributed and Multiprocessor Operating Systems.
Case Studies: Unix/Linux, WINDOWS and other Contemporary Operating Systems.

Course Outcome:

1. Describe basics of computer network, network architecture, TCP/IP protocol suite, OSI reference models & fundamentals of physical layer.
2. Classify data link protocol like flow control, error control, bit-oriented protocol.
3. Paraphrase multi-channel access protocol, IEEE 802 standards & use Ethernet standards.
4. Explain routing &amp; congestion algorithm. State IP protocol, addressing & subnet.
5. Distinguish various transport & application layer protocols.

List of Text/Reference Books:

1. Avi Silber Schatz, Peter Galvin, Greg Gagne, “Operating System Concepts Essentials”, Wiley Asia Student Edition, 10th Edition, 2018.
2. William Stallings, “Operating Systems: Internals and Design Principles”, Prentice Hall of India, 5th Edition, 2005.
3. Charles Crowley, “Operating System: A Design-oriented Approach”, Irwin Publishing, 1st Edition.
4. Gary J. Nutt, “Operating Systems: A Modern Perspective”, Addison-Wesley, 2nd Edition.
5. Maurice Bach, “Design of the Unix Operating Systems”, Prentice-Hall of India, 8th Edition.
6. Daniel P. Bovet, Marco Cesati, “Understanding the Linux Kernel”, O'Reilly and Associates, 3rd Edition.
7. Andrew S. Tanenbaum, “Modern Operating Systems”, Prentice Hall, 3 rdEdition,2007.
8. Bovet & Cesati, “Understanding the Linux Kernel”, O’Reily, 3rd Edition.

List of Experiments:

1. To implement FCFS CPU scheduling algorithm.
2. To implement SJF CPU scheduling algorithm.
3. To implement Priority CPU Scheduling algorithm.
4. To implement Round Robin CPU scheduling algorithm.
5. To compare various CPU Scheduling Algorithms over different Scheduling Criteria.
6. To implement classical inter process communication problem (producer consumer).
7. To implement classical inter process communication problem (Reader Writers).
8. To implement classical inter process communication problem (Dining Philosophers).
9. To implement &compare various page replacement algorithms.
10. To implement & Compare various Disk & Drum scheduling Algorithms
11. To implement Banker’s algorithms.
12. To implement Remote Procedure Call (RPC).
13. Write a Devices Drivers for any Device or peripheral.

IPS Academy, Institute of Engineering & Science

(A UGC Autonomous Institute, Affiliated to RGPV, Bhopal)

Scheme Based on AICTE Flexible Curriculum

Department of Computer Science & Engineering

Bachelor of Technology (B.Tech.) [Computer Science & Engineering (IoT)]

V-Semester

SBC-IO-02	**Programming in Python**	**0L:0T:4P (4hrs.)**	**2 Credits**

Prerequisite: C and C++ language.

Course Objective:
The course is designed to provide Basic knowledge of Python. Python programming is intended for software engineers, system analysts, program managers and user support personnel who wish to learn the Python programming language. Learning Outcomes: Problem solving and programming capability.

Module 1: (04hrs.)
Introduction, History, Features, Python –Environment Setup Local Environment Setup, Getting Python, Installation of Python, Use of IDE.

Module 2: (04hrs.)
Python –Basic Syntax Python Identifiers, Reserved Words, Lines & Indentation, Multiline Statements, Quotation in Python, Comments & other useful constructs, Python –Variables Assigning Values to Variables, Multiple Assignment, Standard Data Types.

Module 3: (04hrs.)
Python –Variables, Assigning Values to Variables, Multiple Assignment, Standard Data Types; Python Numbers, Python Strings, Python Lists, Python Tuples, Dictionary, Data Type Conversion.

Module 4: (04hrs.)
Python –Basic Operators, Types of Operators, Arithmetic Operators, Comparison Operators, Assignment Operators, Bitwise Operators, Logical Operators, Operator Precedence, Python – Decision Making & Loops, Flowchart, If statement Syntax.

Module 5:(06hrs.)
Python-Functions, Syntax for defining a function, Calling a Function, Function Arguments, Anonymous Functions Python-Applications & Further Extensions, Data analysis packages.

Course Outcome:

1. Install Python and have knowledge of syntax of Python.
2. Describe the Numbers, Math functions, Strings, List, Tuples and Dictionaries in Python.
3. Express different decision-Making statements and Functions.
4. Develop code in Python using functions, loops etc.
5. Design GUI Applications in Python and evaluate different database operations.

List of Text/Reference Books:

1. Eric Matthes, "Python Crash Course: A Hands-On, Project-Based Introduction to Programming", No Starch Press.
2. Zed A. Shaw, "Learn Python the Hard Way" (3rd Edition), Addison Wesley.
3. Paul Barry, "Head-First Python", O'Reilly.
4. John Zelle, Franklin, Python Programming, Beedle & Associates Inc.

List of Experiments:

Write a Python program:

1. To find GCD of two numbers.
2. To find the square root of a number by Newton's Method.
3. To find the exponentiation of a number.
4. To find the maximum from a list of numbers.
5. To perform Linear Search.
6. To perform binary search.
7. To perform selection sort.
8. To perform insertion sort.
9. To perform Merge sort.
10. To find first n prime numbers.
11. To multiply matrices.
12. For command line arguments.
13. To find the most frequent words in a text read from a file.
14. To simulate elliptical orbits in Pygame.
15. To bouncing ball in Pygame.
16. To demonstrate data analysis packages using python like Pandas, Filtering, etc.

IPS Academy, Institute of Engineering & Science

(A UGC Autonomous Institute, Affiliated to RGPV, Bhopal)

Scheme Based on AICTE Flexible Curriculum

Department of Computer Science & Engineering

Bachelor of Technology (B.Tech.) [Computer Science & Engineering (IoT)]

V-Semester

PROJ-IO-01	Mini Project	0L:0T:4P (4hrs.)	2 Credits

Prerequisite: A sound knowledge in subjects in respective stream.

Course Objective: The main objective of Mini Project is to offer a platform for students to apply theoretical concepts and programming knowledge in a practical setting, solving real time problems or situations fostering creativity, problem solving abilities and technical proficiencies.

Course Outcome: At the end of the course the student should be able to

1. Understand, plan, and execute a Mini Project with team.
2. Acquire knowledge within the chosen area of technology for project development.
3. Identify, discuss, and justify the technical aspects of the chosen project with a comprehensive and systematic approach.
4. Communicate and report effectively project related activities and findings.
5. Formulate and propose a plan for creating a solution for the problems identified.
6. Report and present the finding of the study conducted in the preferred domain.

Guidelines:

The mini project is desirable to be done in a group of 2 students. Each group has to prepare a title related to any engineering discipline, and the title must emulate any real-world problem.