PC401CS **COMPUTER ORGANIZATION**

Credits: 3

Instruction: (3L+1T) hrs per week *Duration of SEE: 3 hours*
CIE: 30 marks *SEE: 70 marks*

Course Objectives:

- To understand the Instruction Set Architecture: Instruction format, types, various addressing modes
- To understand the basic components and design of the CPU: the ALU and control unit
- To understand the parallelism both in terms of a single processor and multiple processors
- To understand the I/O Organization, Interrupt-driven I/O, and DMA

Course Outcomes:

Student will be

- Able to understand the Instruction Set Architecture: Instruction format, types, various addressing modes
- Able to understand the basic components and design of the CPU: the ALU and control unit write multi threaded programs with synchronization.
- Able to understand the parallelism both in terms of a single processor and multiple processors
- Able to understand the I/O Organization, Interrupt-driven I/O, DMA

UNIT -I

Data Representation: Data types, Complements, Fixed and Floating Point representations, and Binary codes.
Overview of Computer Function and Interconnections: Computer components, Interconnection structures, Bus interconnection, Bus structure, and Data transfer.

UNIT-II

Register Transfer Microoperations: Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic, Logic and Shift microperations, Arithmetic Logic Shift Unit.
Basic Computer Organization and Design: Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory reference instruction, Input-Output and Interrupt.

UNIT-III

Microprogrammed Control: Control memory, Address Sequencing, Microprogram example, Design of Control Unit.
Central Processing Unit: General Register Organization, Stack Organization, Instruction formats, Addressing modes, Data Transfer and Manipulation, and Program control.

Computer Arithmetic: Addition and Subtraction, Multiplication, Division, and Floating Point Arithmetic Operations.

UNIT-IV

Memory Organization: Memory Hierarchy, Main Memory, RAM and ROM, Auxiliary memory, Associative memory, Cache memory, Virtual memory, Memory Management hardware.

UNIT–V

Input-Output Organization: Peripheral Devices, Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt, Direct Memory Access (DMA), I/O Processor, Serial Communication.
Pipeline Processing: Arithmetic, Instruction and RISC Pipelines.
Assessing and Understanding Performance: CPU performance and its factors, Evaluating performance.

Suggested Reading:

1. Morris Mano M, *Computer System Architecture*, Pearson Education India, 3rd Edition, 2007.
2. William Stallings, *Computer Organization and Architecture*, PHI, 7th Edition, 2008.
3. David A Patterson, John L Hennessy, *Computer Organization and Design*, Morgan Kaufmann, 5th Edition, 2013.

PC402 CS OBJECT ORIENTED PROGRAMMING USING JAVA

Credits: 3

Instruction: (3L+1T) hrs per week *Duration of SEE: 3 hours*
CIE: 30 marks *SEE: 70 marks*

Course Objectives:

- To introduce fundamental object oriented concepts of Java programming Language - such as classes, inheritance packages and interfaces.
- To introduce concepts of exception handling and multi threading.
- To use various classes and interfaces in java collection framework and utility classes.
- To understand the concepts of GUI programming using AWT controls.
- To introduce Java I/O streams and serialization.

Course Outcomes:

Student will be

- Able to understand the usage of abstract classes.
- Able to write multi threaded programs with synchronization.
- Able to implement real world applications using java collection frame work and I/O classes
- Able to write Event driven GUI programs using AWT/Swing

UNIT -I

Object Oriented System Development: understanding object oriented development, understanding object oriented concepts, benefits of object oriented development**.**
Java Programming Fundamentals: Introduction, overview of Java, data types, variables and arrays, operators, control statements, classes, methods, inheritance, packages and interfaces**.**

UNIT -II

Exceptional Handling, Multithreaded Programming, I/O Basics, Reading Console Input and Output, Reading and Writing Files, Print Writer Class, String Handling.

UNIT -III

Exploring Java.Lang, Collections Overview, Collection Interfaces, Collection Classes, Iterators, Random Access Interface, Maps, Comparators, Arrays, Legacy Classes and Interfaces, String Tokenizer, Bitset, Date, Calendar, Observable Timer.

UNIT –IV

Introducing AWT working With Graphics: AWT Classes, Working with Graphics.

Event Handling: Two Event Handling Mechanisms, The Delegation Event Model, Event Classes, Source of Events, Event Listener Interfaces.

AWT Controls: Control Fundamentals, Labels, Using Buttons, Applying Check Boxes, CheckboxGroup, Choice Controls, Using Lists, Managing Scroll Bars, Using TextField, Using TextArea, Understanding Layout Managers, Menu bars and Menus, Dialog Boxes, FileDialog, Handling events by Extending AWT Components, Exploring the controls, Menus and Layout Managers.

UNIT –V

Java I/O Classes and Interfaces, Files, Stream and Byte Classes, Character Streams, Serialization.

Suggested Reading:

1. Herbert Schildt, *The Complete Reference JAVA*, Tata McGraw Hill, 7thEdition, 2005.
2. James M Slack, *Programming and Problem Solving with JAVA*, Thomson Learning, 2002.
3. C.Thomas Wu, *An Introduction to Object-Oriented Programming with Java*, Tata McGraw Hill, 5th Edition, 2005.

PC403CS **PROGRAMMING LANGUAGES**

Credits: 3

Instruction: (3L+1T) hrs per week *Duration of SEE: 3 hours*
CIE: 30 marks *SEE: 70 marks*

Course Objectives:

- To introduce the major programming paradigms, and the principles and techniques involved in design and implementation of modern programming languages.
- To introduce notations to describe syntax and semantics of programming languages.
- To analyze and explain behavior of simple programs in imperative languages using concepts such as binding, scope, control structures, subprograms and parameter passing mechanisms.
- To introduce the concepts of ADT and object oriented programming for large scale software development.
- To introduce the concepts of concurrency control and exception handling.

Course Outcomes:

Students will be

- Able to understand the programming paradigms of modern programming languages.
- Able to introduce notations to describe syntax and semantics of programming languages.
- Able to analyze the behavior of simple programs in imperative languages.
- Able to understand the concepts of ADT and object oriented programming for large scale software development.

UNIT-I

Preliminary Concepts: Reasons for Studying Concepts of Programming Languages, Programming Domains, Language Evaluation Criteria, Influences on Language Design, Language Categories, Language Design Trade-offs, Implementation Methods, Programming Environments, Evolution of the Major Programming Languages.

Describing Syntax and Semantics: General Problem of Describing Syntax, Formal Methods of Describing Syntax, Attribute Grammars, Describing the Meaning of Programs.

UNIT-II

Names, Binding, Type Checking, and Scopes: Names, Variables, The Concept of Binding, Type Checking, Strong Typing, Type Compatibility, Scope, Scope and Lifetime, Referencing Environments, Named Constants.

Data Types: Primitive Data Types, Character String Types, User- Defined Ordinal Types, Array Types , Associative Arrays, Record Types, Union Types, Pointer and Reference Types.

Expressions and Assignment Statements: Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short-Circuit Evaluation, Assignment Statements, Mixed- Mode Assignment.

UNIT-III

Statement-Level Control Structures: Selection Statements, Iterative Statements, Unconditional Branching, Guarded Commands.
Subprograms: Fundamentals and Design Issues for Subprograms, Local Referencing Environments, Parameter –Passing Methods, Parameters That are Subprograms Names, Overloaded Subprograms, Generic Subprograms, Design Issues for Functions, User-Defined Overloaded Operators.
Implementing Subprograms: The General Semantics of Calls and Returns, Implementing "Simple" Subprograms, Implementing Subprograms with Stack-Dynamic Local Variables, Nested Subprograms, Blocks, Implementing Dynamic Scoping.
Abstract Data Types: The Concept of Abstraction, Introduction to Data Abstraction, Design Issues for Abstract Data Types, Language Examples, Parameterized ADT, Encapsulation Constructs, Naming Encapsulation.

UNIT-IV
Object Oriented Programming: Design Issues, Object Oriented Programming in Smalltalk, C++, Java, C#, Ada 95, Ruby, The Object Model of JavaScript, Implementation of Object Oriented Constructs.
Concurrency: Subprogram level Concurrency, Semaphores, Monitors, Massage Passing, Ada Support for Concurrency, Java Threads, C# Threads, Statement-Level Concurrency.
Exception Handling and Event Handling: Introduction to Exception Handling, Exception Handling in Ada, C++ and Java, Introduction to Event Handling, Event Handling with Java.

UNIT- V
Functional Programming Languages: Introduction, Mathematical Functions, Fundamentals of FPL, LISP, Introduction to Scheme, COMMON LISP, ML, Haskell, Application of Functional Programming Languages and A Comparison of Functional and Imperative Languages.
Logic Programming Languages: Introduction to Predicate Calculus, Predicate Calculus and Proving Theorems, An Overview of Logic Programming. The Origins, Basic Elements and Deficiencies of Prolog, Applications of Logic Programming.
Scripting Languages: Key concepts, Case Study: Python(From the Suggested Reading 2).

Suggested Reading:

1. Robert .W. Sebesta, *Concepts of Programming Languages* , Pearson Education, 8th Edition, 2008.
2. Watt, *Programming Languages* , Wiley Dreamtech, 1st Edition, 2004.
3. Louden , Programming *Languages*, Cengage, 2nd Edition, 2003.
4. Ghezzi , *Programming Languages*, John Wiley, 3rd Edition, 1998.
5. Pratt and Zelkowitz, *Programming Languages Design and Implementation* , PHI/Pearson Education, 4th Edition ,2001.

PC404CS MICROPROCESSORS AND INTERFACING

Credits: 3

Instruction: (3L+1T) hrs per week *Duration of SEE: 3 hours*
CIE: 30 marks *SEE: 70 marks*

Course Objectives:

- Able to understand the architecture and organization of microprocessor along with instruction coding formats.
- Able to understand, write programs in assembly language
- Able to understand the memory and addressing concepts for interfacing I/O devices to the microprocessor.
- Able to understand software/ hardware interrupts and further write programs to perform I/O using handshaking and interrupts
- Able to have an understanding of digital interfacing and system connections.

Course Outcomes:

Students will be

- Able to understand the architecture and organization of microprocessor.
- Able to write programs in assembly language.
- Able to apply knowledge for interfacing I/O devices to the microprocessor.
- Able to understand software/ hardware interfacing and system connections.

UNIT –I

8085 Architecture: Introduction to microprocessors and microcontrollers, 8085 Processor Architecture, Internal operations, Instructions and timings. Programming the 8085 - Introduction to 8085 instructions, Addressing modes and Programming techniques with Additional instructions.

UNIT-II

Stacks and Subroutines, Interfacing Peripherals - Basic Interfacing concepts, Interfacing output displays, Interfacing input keyboards. Interrupts - 8085 Interrupts, Programmable Interrupt Controller (8259A). Direct Memory Access (DMA) - DMA Controller (Intel 8257), Interfacing 8085 with Digital-to-Analog and Analog-to-Digital converters.

UNIT-III

Programmable Peripheral Interface (Intel 8255A), Programmable Communication Interface (Intel 8251), Programmable interval timer (Intel 8253 and 8254), Programmable Keyboard/ Display controller (Intel 8279). Serial and Parallel Bus standards RS 232 C, IEEE 488.

UNIT-IV

Introduction to Microcontrollers: 8051 – Architecture, Instruction set, Addressing modes and Programming techniques. Comparison of various families of 8-bit micro controllers. System Design Techniques - Interfacing of LCD, ADC, Sensors, Stepper motor, Keyboard and DAC using microcontrollers.

UNIT –V

Applications of Microcontrollers: Interfacing Keyboards, Interfacing LCD, Interfacing LED, Interfacing ADC&DAC, Interfacing with Sensors, RTC, Stepper Motor.
Communication standards - Serial RS 232 and USB.

Suggested Reading:

1. Ramesh S. Gaonkar, *Microprocessor Architecture, Programming, and Applications with the 8085*, Prentice Hall India, 5th Edition , 2002.
2. Kenneth Ayala, *The 8051 Microcontroller*, Cengage Learning, 3rd Edition, 2007.
3. Muhammed Ali Mazidi, *The 8051 Microcontroller & Embedded Systems*, Pearson Education, India, 2nd Edition, 2007.
4. Myke Predko, *Programming and Customizing the 8051 Microcontroller*, Tata McGraw Hill Education, 2000.

BS402MT **MATHEMATICS AND STATISTICS**

Credits: 3

Instruction: (3L+1T) hrs per week *Duration of SEE: 3 hours*
CIE: 30 marks *SEE: 70 marks*

Course Objectives:

- To introduce transforms like Laplace, Fourier and to study their properties
- To introduce number theory and its applications
- To provide the knowledge of probability distributions like uniform, normal and exponential distributions, Tests of significance, correlation and regression.

Course Outcomes:
Students will be

- Able to solve differential equations using Laplace and Fourier transforms
- Able to solve problems in elementary number theory
- Able to apply various probability distributions to solve practical problems, to estimate unknown parameters of populations and apply the tests of hypotheses.
- Able to perform a regression analysis and to compute and interpret the coefficient of correlation

UNIT- I
Laplace Transforms: Introduction of Laplace transforms, sufficient condition for existence of Laplace transform, Laplace transform of Derivatives, Laplace transform of integrals, Translation theorems (I & II shifting theorems), Differentiation of Laplace transform (Multyplication by t), Integration of Laplace transform (Division by t), convolution theorem, Solving initial value problems using Laplace transform.

UNIT- II
Fourier Transforms: Introduction, Fourier integrals, Fourier sine and cosine integrals, Complex form of Fourier integral, Fourier transform, Fourier sine and cosine transforms, Finite Fourier sine and cosine transforms, Properties of Fourier transforms, Convolution theorem for Fourier transforms.

UNIT- III

Number Theory: Divisibility and Modular arithmetic, integer representation, primes and GCD, solving congruences and applications, Introduction to cryptography.

UNIT- IV

Probability: Random variables, Uniform, Normal, Exponential distributions, Mean, median, mode and standard deviation, Conditional probability and Baye's theorem, Tests of significance, t-test, F-test and χ^2 test.

UNIT- V

Curve Fitting: Curve fitting by method of least squares, correlation and regression, types of correlations, Karl Pearson's coefficient of correlation, Spearman's rank correlation coefficient, equal ranks, equations to the lines of regression.

Suggested Reading:

1. R.K.Jain & S.R.K Iyengar, *Advanced Engineering Mathematics*, Narosa Publication, 4th Edition, 2014.
2. B.S.Grewal, *Higher Engineering Mathematics*, Khanna Publications, 43rd Edition, 2014.
3. Vasishtha and Gupta, *Integral Transforms*, Krishnan Prakashan Publications, 2014.
4. Erwin Kreyszig, *Advanced Engineering Mathematics*, John Wiley &Sons, 10th Edition, 2012.
5. James S.Kraft and Lawrence C.Washington, *An Introduction to Number Theory and Cryptography*, CRC press, 2016.

ES421EC **SIGNALS AND SYSTEMS**

Credits: 3

Instruction: (3L+1T) hrs per week *Duration of SEE: 3 hours*
CIE: 30 marks *SEE: 70 marks*

Course Objectives:

- To learn basic concepts related to signals & systems.
- To familiarize with basic operations on signals mathematical representation of periodic, aperiodic signals continuous discrete systems.
- To understand convolution, correlation operations on continuous signals.
- To analyze the response of systems on application of step, ramp inputs using Fourier & Z transforms.

Course Outcomes:
Students will be

- Able to differentiate signal like discrete time, continuous time, power, energy, periodic, aperiodic, even, odd.
- Able to define the system by an impulse response with properties: memoryless, causal, stable.
- Able to understand the properties of FT, Z-transform & LT.

UNIT- I

Signal Analysis: Analogy between vectors and signals, Orthogonal signal space, Signal approximation using orthogonal functions, Mean square error, Closed or complete set of orthogonal functions, Orthogonality in complex functions, Exponential and sinusoidal signals, Concepts of Impulse function, Unit step function, Signum function.

UNIT-II

Fourier Transform: Deriving Fourier transform from Fourier series, Fourier transform of arbitrary signal, Fourier transform of standard signals, Fourier transform of periodic signals, properties of Fourier transforms, Fourier transforms involving impulse function and Signum function. Introduction to Hilbert Transform.

UNIT-III

Signal Transmission Through Linear Systems: Linear system, impulse response, Response of a linear system, Linear time invariant (LTI) system, Linear time variant (LTV) system, Transfer function of a LTI system. Filter characteristics of linear systems. Distortion less transmission through a system, Signal bandwidth, system bandwidth, Ideal LPF, HPF and BPF characteristics, Causality and Poly-Wiener criterion for physical realization, relationship between bandwidth and rise time.

.

UNIT–IV

Convolution & Correlation of Signals: Concept of convolution in time domain and frequency domain, Graphical representation of convolution, Convolution property of Fourier transforms. Cross correlation and auto correlation of functions, properties of correlation function, Energy density spectrum, Parseval's theorem, Power density spectrum, Relation between auto correlation function and energy/power spectral density function.Relation between convolution and correlation, Detection of periodic signals in the presence of noise by correlation, Extraction of signal from noise by filtering.

UNIT- V

Z-Transform: Fundamental difference between continuous and discrete time signals, discrete time signal representation using complex exponential and sinusoidal components, Periodicity of discrete time usingcomplex exponential signal, Concept of Z- Transform of a discrete sequence. Distinction between Laplace, Fourier and Z transforms. Region of convergence in Z-Transform, constraints on ROC for various classes of signals, Inverse Z- transform, properties of Z-transforms.

Suggested Reading:

1. Lathi B.P., *Signals Systems & Communications*, B.S. Publications, 1st Edition, 2006.
2. Alan V. Oppenheim, Alan.S.Willsky, S Hamid Nawab, *Signals Systems*, Prentice Hall of India, 2nd Edition, 2007.
3. Simon Haykin and Van Veen, *Signals and Systems* , Wiley India, 2nd Edition, 2008.

PC451CS **JAVA PROGRAMMING LAB**

Credits: 1

Instruction: (2P) hrs per week *Duration of SEE: 3 hours*
CIE: 25 marks *SEE: 50 marks*

Course Objectives:

- Write programs using classes, inheritance and abstract classes.
- Write multi threaded programs with synchronization.
- Write real world applications using java collection frame work and I/O classes
- Write Event driven GUI programs using AWT/Swing

Course Outcomes:

Student will be

- Able to understand the usage of abstract classes.
- Able to write multi threaded programs with synchronization.
- Able to implement real world applications using java collection frame work and I/O classes
- Able to write Event driven GUI programs using AWT/Swing

1. A program to illustrate the concept of class with constructors, methods and overloading.
2. A program to illustrate the concept of Inheritance and Dynamic polymorphism.
3. A program to illustrate the usage of Abstract class.
4. A program to illustrate Multithreading.
5. A program to illustrate Thread synchronization.
6. A program using String tokenizer.
7. A program using Linked list class.
8. A program using Tree set class.
9. A program using Hash set and Iterator classes.
10. A program using Map classes.
11. A program using Enumeration and Comparator interfaces.
12. A program to illustrate the usage of Filter and Buffered I/O streams.
13. A program to illustrate the usage of Serialization.
14. An application involving GUI with different controls, menus and event handling.
15. A program to implement AWT/Swing.

PC452CS **MICROPROCESSORS LAB**

Credits: 1

Instruction: (2P) hrs per week *Duration of SEE: 3 hours*
CIE: 25 marks *SEE: 50 marks*

Course Objectives:

- Write simple assembly language program using 8085 instruction set
- Write programs to interface various peripheral devices with 8085.
- Design simple applications using 8051 Micro controller.

Course Outcomes:

Students will be

- Able to write assembly language program using 8085 instruction set
- Able to apply programming skills for interfacing various peripheral devices with 8085.
- Able to develop applications using 8051 Micro controller.

PART A: **8085 Programming using Microprocessor Trainer Kit**

1. Simple programming examples using 8085 instruction set. To understand the use of various instructions and addressing modes.
2. Interfacing and programming of 8255. (e.g.: traffic light controller)
3. Interfacing and programming of 8254.
4. Interfacing and programming of 8279.

PART B: 8051 Programming

1. Simple programming examples using 8051 Microcontroller
2. A/D and D/A Converter Interface
3. Stepper motor interface
4. Display interface

MC461CS SOCIETY OUTREACH PROGRAM

Credits: 2 Units

Instruction: (2P) hrs per week
CIE: 50 marks

Course Objectives:

- To prepare the students to sensitize the society on social issues, particularly on environment, health and literacy.
- To prepare the students to learn how to pay back to the society and the Nation.
- To change the attitude of the society.
-

Course Outcomes:

Students will be

- Able to find the scientific solutions for a specific problem in the society.
- Able to demonstrate the leadership qualities.
- Able to bring out their latent talent
- Able to develop interpersonal skills and team-spirit.

The Society Outreach Program gives students the opportunity to understand and involve in community service. Projects are student-led and may encompass the activities related to the following:

Environmental issues,

Health education for the community and sanitation,

Communal harmony and peace education,

Legal Awareness/Rights,

Human rights and rights of vulnerable groups;

Panchayats and development issues;

Women's Empowerment and

Social issues and gender issues.

Awareness on Solid waste management and any possible solutions;

Awareness on natural resources, solar energy and wind energy;

Awareness of work culture, punctuality, discipline, cleanliness in house and surroundings, and Intake of quality food;

The students are required to make a study and/or participate in society outreach programmes and submit a report.

The department will appoint a project coordinator who will be incharge of the following:

- Grouping of students (a maximum of three in group)
- Allotment of project guides
- Project monitoring at regular intervals

The students need to make a presentation on the issue they have worked/studied. The work carried out, the report and the presentation carry 50 marks.

PW461CS **MINI PROJECT**

Credits: 2

Instruction: (2P) hrs per week *Duration of SEE: 3 hours*
CIE: 25 marks *SEE: 50 marks*

The students are required to carry out mini project that involves usage of C/C++/JAVA/Microprocessors/Microcontroller Programming, etc.,

The emphasis will be on the usage of various data structures.

The department will appoint a project coordinator who will be incharge of the following:

- Grouping of students (a maximum of three in group)
- Allotment of project guides
- Project monitoring at regular intervals

All the projects are to be evaluated by a monitoring committee comprising of project coordinator and the supervisor on the basis of an oral presentation, demonstration, mini project report and Viva-Voce.