Purbanchal University

Bachelor in Information Technology (BIT)

Year: II **Semester: III**

S.N	Course Code	Course description	Credits	Lecture (Hrs)	Tutorial (Hrs)	Practical (Hrs)	Total (Hrs)
1	BIT270CO	System Analysis & Design	3	3	1	-	4
2	BIT272CO	Microprocessor & Assembly Language	3	3	1	2	6
3	BIT273CO	Data Structure & Algorithm	3	3	1	2	6
4	BIT275CO	User Interface Design	3	3	1	2	6
5	BIT278CO	Project-III	2	-	-1	4	4
6	BIT280CO	Numerical Methods	3	3	1	2	6
		Total	**17**	**15**	**5**	**13**	**32**

System Analysis & Design

BIT270CO

Year: II **Semester: III**

Teaching Schedule Hours/Week			Examination Scheme				
Theory	Tutorial	Practical	Internal Assessment		Final		Total
			Theory	Practical	Theory	Practical	
3	1	-	20	-	80	-	100

Course Objective

This course helps to launch the careers of successful system analysts or of users assuming an active role in building systems that satisfy their organizations' information needs. The course also provides a solid foundation of systems.

Course contents

1. **Overview of system analysis and design** **[6 Hrs]**
   - 1.1 Introduction to system analysis and design
   - 1.2 Information systems and its types
   - 1.3 Stakeholders of information systems
   - 1.4 Systems development life cycle and life cycle models (waterfall, spiral, prototype)
   - 1.5 Introduction to CASE tools

2. **Process and conceptual modeling** **[8 Hrs]**
   - 2.1 Introduction to data flow diagram (DFD)
   - 2.2 Concepts used in drawings DFDs
   - 2.3 DFD design (up to level 2)
   - 2.4 Conceptual modeling
   - 2.5 Entity relationship diagrams

3. **Logic modeling** **[3 Hrs]**
   - 3.1 Decision table
   - 3.2 Decision tree
   - 3.3 Structured English
   - 3.4 Data dictionary

4. **Systems analysis** **[8 Hrs]**
   - 4.1 System planning and initial investigation
   - 4.2 Project scheduling

4.3 Requirement analysis
4.4 Types of requirements
4.5 Requirement gathering methods
4.6 Feasibility study and its types
4.7 Steps of feasibility study
4.8 Cost/benefits analysis (payback method, NPV method)

5. **Systems design** **[8 Hrs]**
5.1 Introduction to system design
5.2 The process and stages of system design
5.3 Logical and physical design
5.4 Introduction to structured design (modular system design, functional strength, structure chart, cohesion, coupling)
5.5 Database design and overview of file organization
5.6 Input/output and forms design

6. **System implementation** **[6 Hrs]**
6.1 Introduction to system implementation
6.2 System installation and its types
6.3 System quality, software quality assurance (formal technical review, walkthrough, inspections)
6.4 System maintenance, types of maintenance and process of system maintenance
6.5 Introduction to system testing

7. **Object-oriented analysis and design** **[6 Hrs]**
7.1 Object-oriented development life cycle
7.2 The unified modeling language
7.3 Use-case modeling
7.4 Object modeling: class diagrams
7.5 Dynamic modeling: state diagrams
7.6 Dynamic modeling: sequence diagrams

Reference books

- Jeffery A. Hoffer, Joey F. George, Joseph S. Valacich, "*Modern System Analysis & Design*", Pearson Education, 2nd edition
- "*Introduction to System Analysis & Design*", Igor Hawrysjkiewycz, PHI, 4th edition
- Englewood Cliffs, New Jersey, "*System Analysis & Design*".
- Jeffrey L. Whitten, Loonnie D. Bentley, "*System Analysis & Design Methods*", 5th edition
- Grady Booch, "*Object Oriented Analysis & Design with Application*", Pearson eduction

Microprocessor & Assembly Language

BIT272CO

Year: II **Semester: III**

Teaching Schedule Hours/Week			Examination Scheme				
Theory	Tutorial	Practical	Internal Assessment		Final		Total
3	1	2	Theory	Practical	Theory	Practical	150
			20	50	80	-	

Course Objective

To be familiar with the operation, programming and application of 8 and 16 bits microprocessor.

Course contents

1. **Introduction** **[5 Hrs]**
   - 1.1 History of microprocessor
   - 1.2 Calculator and stored program computer
   - 1.3 Von Neumann and Harvard architecture
   - 1.4 Simple stored program computer architecture
   - 1.5 Applications of microprocessor

2. **Intel 8085 microprocessor** **[10 Hrs]**
   - 2.1 Pin diagram and pin functions
   - 2.2 Internal architecture
   - 2.3 Addressing modes
   - 2.4 Instruction set with classification
   - 2.5 Instruction format and programming
   - 2.6 Fetch and execution cycle
   - 2.7 Fetch execution overlap
   - 2.8 Timing diagram

3. **Bus structure and memory devices** **[4 Hrs]**
   - 3.1 Bus structure, synchronous and asynchronous data bus, address bus, bus timing
   - 3.2 Memory devices
   - 3.3 Static and dynamic RAM, ROM
   - 3.4 Address decoding, memory interface (8, 16, 32, 64 bits)

4. **Input/output interfaces** **[6 Hrs]**
   - 4.1 Serial communication
     - 4.1.1 Asynchronous and synchronous interface

4.1.2 8251 programmable communication interface (block diagram and modes only)

4.2 Parallel communication

4.2.1 8255 Programmable peripheral interface (block diagram and modes only)

4.3 RS-232 and IEEE 488-1978 general purpose interface standard

4.4 Keyboard and display controller (block diagram only)

5. **Interrupt (8 and 16 bits)** **[5 Hrs]**

5.1 Introduction

5.2 Basic interrupt processing

5.3 Types of interrupts

5.4 Interrupt service routing requirements

5.5 Interrupt priority

6. **DMA** **[3 Hrs]**

6.1 Introduction

6.2 Basic DMA operation

6.3 8237 DMA controller (block diagram and modes only)

7. **8086 instruction description and assembler directives** **[10 Hrs]**

7.1 Pin diagram and pin function

7.2 Internal architecture

7.3 Addressing modes

7.4 Assembler instruction format: opcodes, mnemonics and operands

7.5 Assembler operations: sample assembly language program and code generation, one-pass and two-pass assembly, assembler directives

7.6 Instruction set with classification and programming

8. **Introduction to higher series of Intel processors, A comparative study** **[2 Hrs]**

Laboratory

There shall be following laboratory exercises using the microprocessor trainer-kit and assembler.

- Familiarization with 8085 and 8086 microprocessor trainer-kit and their simulators
- Data transfer, arithmetic and logical instructions
- Subroutines and branching instructions
- Stack operation
- Timers and delay
- Code conversion

Reference books

- Ramesh S. Gaonkar, "Microprocessor – Architecture, Programming & Applications with 8085", Penram Intl. publisher, 5th edition, 2006.
- Ghose P. K., Sridhar P. R. "0000 to 8085: Introduction to Microprocessor for Engineers & Scienists", 2nd edition, PHI
- Barry B. Berry, "The Intel Microprocessor 8086, 8088, 80186, 80286, 80386 & 80486 (Architecture, Programming & Interface)", PHI
- Lance, A. Leventhal, "Introduction to Microprocessors: Software, Hardware & Programming", Eastern Economy Edition. PHI
- Yu Cheng Liu & Glenn A. Gibson, "Microprocessor Systems: The 8086/8088 Family", PHI
- Douglas V. Hall, "Microprocessors & Interfacing", PHI

Chapter wise marks distribution for questions:

Unit	Group A: Long Type (Attempt any 2 out of 3)	Group B: Short Type (Attempt any 7 out of 8-9)
1	-	1
2	1	2
3	-	1
4	1	1
5	1	1
6	-	1
7	1	2
8	-	1
	2 × 12 = 24	7 × 8 = 56

Data Structure & Algorithm

BIT273CO

Year: II **Semester: III**

Teaching Schedule Hours/Week			Examination Scheme				
Theory	Tutorial	Practical	Internal Assessment		Final		Total
3	1	2	Theory	Practical	Theory	Practical	150
			20	50	80	-	

Course Objective

To provide fundamental knowledge of data structure, various algorithms used and their implementations.

Course contents

1. **Introduction** **[2 Hrs]**
   - 1.1 Data and data types
   - 1.2 Data structure and its operations and importance
   - 1.3 ADT and its applications and importance
   - 1.4 ADT vs DS

2. **Algorithmic efficiency and its complexity** **[2 Hrs]**
   - 2.1 Time and space analysis
   - 2.2 Asymptotic notations – big O, big sigma, theta, omega

3. **Stack** **[4 Hrs]**
   - 3.1 Definition
   - 3.2 Primitive operations with examples representing stack in C
   - 3.3 Stack implementation (PUSH/POP) operations
   - 3.4 Stack as an ADT
   - 3.5 Prefix, infix and postfix expressions
     - 3.5.1 Definitions
     - 3.5.2 Algorithms for evaluation of infix and postfix expression
     - 3.5.3 Converting an expression from infix to postfix and vice versa

4. **Queue** **[3 Hrs]**
   - 4.1 Definition
   - 4.2 Primitive operations with examples representing queue in C
   - 4.3 Queue implementation (Enqueue/Dequeue) operations
   - 4.4 Queue as an ADT

4.5 Types of queue (linear, circular, priority queue and its types)

5. **List and linked list** **[6 Hrs]**

5.1 Introduction to list and linked list

5.2 Advantages of list over stack and queue

5.3 Types of lists (static and dynamic)

5.4 List and list operations

5.5 Array implementation of list

5.6 Linked list as an ADT

5.7 Linked list and its types

5.7.1 Linear linked list – singly linear and doubly linear

5.7.2 Circular linked list – singly circular and doubly circular

5.8 Linked list operations (insertion/deletion from the front node, from the last node, before a given node, after a given node)

5.9 Linked stack and linked queue

5.10 Doubly linked list and its advantages

6. **Recursion** **[4 Hrs]**

6.1 Definition and recursive functions

6.2 Recursion vs iteration with advantages and disadvantages

6.3 Application of recursion – factorial calculation, Fibonacci series, TOH, natural numbers multiplication with algorithms and examples

6.4 Efficiency of recursion

7. **Trees** **[6 Hrs]**

7.1 Concepts and definitions

7.2 Binary tree and its applications

7.3 Basic operations in binary tree – insertion/deletion, traversing

7.4 Binary tree traversals – pre-order, post-order and in-order

7.5 Height, depth and level of binary tree

7.6 Balanced trees and balancing algorithms (AVL balanced tree, Huffman coding)

8. **Sorting** **[5 Hrs]**

8.1 Definition and types of sorting (internal and external sort, insertion and selection sort, exchange/bubble sort, quick sort, merge sort, radix sort, shell sort, heap and heap sort)

8.2 Efficiency of sorting

9. **Searching and hashing** **[6 Hrs]**

9.1 Definition of search and concepts of keys, essentials of searching

9.2 Types of searching – sequential search, binary search, binary tree search

9.3 General search tree

9.4 Definition of hashing

9.5 Hash functions and hash table

9.6 Collision resolution technique

9.7 Efficiency comparisons of different search techniques

10. **Graphs** **[7 hrs]**

10.1 Definition and representation of graphs

10.2 Application of graphs

10.3 Graphs as an ADT

10.4 Adjacency matrix implementation, transitive closure, Warshall's algorithm

10.5 Types of graphs

10.6 Graph traversal – depth first search (DFS), breadth first search (BFS)

10.7 Spanning tree and spanning forest

10.8 Kruskal's algorithm, Round-Robin algorithm, Greedy algorithm, Dijkstra's algorithm

Laboratory

There shall be following lab exercises based on C or C++.

- Implementation of stack
- Implementation of linear and circular queues
- Solution of TOH and Fibonacci Recursion
- Implementation of linked list: singly and doubly linear and circular linked list
- Implementation of trees: AVL trees, balancing of AVL
- Implementation of merge sort
- Implementation of search: sequential, tree and binary
- Implementation of graphs: graph traversals
- Implementation of hashing
- Implementation of heap

Reference books

- "*Data Structure using C & C++*", Aarton M. Tenenbaum, Y. Langsam, M. J. Augenstein, PHI.
- "*Fundamental of Computer Algorithms*", H. Sahani
- "*Data Structure of Program Design in C*", Robert L. Kruse, B. P. Leung, C. L. TOndo, PHI
- "*The Art of Programming, Sorting & Searching*", Donald E. Knuti-I.
- "*Data Structure & Application*", Trebly & Sorenson
- "*Introduction to Data Structure & Algorithms with C & C++*", G. W. Rowe, PHI
- "*Fundamentals of Algorithms*", G. Brassand & P. Bratley, PHI

User Interface Design

BIT275CO

Year: II **Semester: III**

Teaching Schedule Hours/Week			Examination Scheme				
Theory	Tutorial	Practical	Internal Assessment		Final		Total
3	1	2	Theory	Practical	Theory	Practical	150
			20	50	80	-	

Course Objective

To provide the design knowledge of use interface and its environment.

Course contents

1. **The goal** **[8 Hrs]**
   - 1.1 Goal directed design
     - 1.1.1 User's goals
     - 1.1.2 Features of user interface design
   - 1.2 Software design
     - 1.2.1 Introduction
     - 1.2.2 Software design vs interface design
   - 1.3 Models of interface design
     - 1.3.1 Conceptual model
     - 1.3.2 Implementation model
     - 1.3.3 Manifest model
     - 1.3.4 Modeling from user's point of view
   - 1.4 Visual interface design
     - 1.4.1 Visual patterns
     - 1.4.2 The canonical vocabulary

2. **The form** **[8 hrs]**
   - 2.1 Interface paradigms
     - 2.1.1 Metaphor
     - 2.1.2 Idioms and branding
     - 2.1.3 Affordances
   - 2.2 Child forms
     - 2.2.1 Usage of window space
     - 2.2.2 Windows pollution
   - 2.3 File system
     - 2.3.1 Introduction

2.3.2 Unified file model

2.3.3 Document management

2.3.4 Storage and retrieval

2.4 Platform independence

2.4.1 Development platform

2.4.2 Multi-platform development

2.4.3 Inter-operability

3. **Software behavior** **[6 Hrs]**

3.1 Flow

3.1.1 Sensible interaction

3.1.2 Flow of states

3.1.3 Notion of MDI states

3.2 Overhead

3.2.1 Revenue tasks and excise tasks

3.2.2 Eliminating excise tasks

3.3 Task coherence

3.3.1 Decision-set streamline

3.3.2 Preference threshold

4. **User-computer interaction** **[8 Hrs]**

4.1 Mouse

4.1.1 Indirect manipulation

4.1.2 Mouse events

4.1.3 Focus and cursor hinting

4.2 Selection

4.2.1 Indicating selection

4.2.2 Insertion and replacement

4.2.3 Additive selection

4.2.4 Group selection

4.3 Gizmos manipulation

4.3.1 Repositioning

4.3.2 Resizing and reshaping

4.3.3 Visual feedback of manipulation

4.4 Drag and drop

4.4.1 Source and target

4.4.2 Problems and solutions

4.4.3 Drag and drop mechanisms

5. **The cast** **[8 Hrs]**

5.1 Menu design issues

5.1.1 Hierarchy of menus

5.1.2 Drop down menus

5.1.3 Pop up menus

5.2 Menus and its types

5.2.1 Standard menus

5.2.2 Optimal menus

5.2.3 System menu

5.2.4 Menu item variation

5.3 Dialog boxes

5.3.1 Dialog box basics

5.3.2 Suspension of interaction

5.3.3 Modal and modeless dialog boxes

5.3.4 Problems in modeless dialog boxes

5.3.5 Different types of dialog boxes

5.4 Dialog box conventions

5.4.1 Caption bar

5.4.2 Attributes

5.4.3 Terminating dialog box

5.4.4 Expanding dialog box

5.4.5 Cascading dialog box

5.5 Toolbars

5.5.1 Advantages over menus

5.5.2 Monetary button and latching button

5.5.3 Customizing toolbars

6. **The Gizmos** **[7 Hrs]**

6.1 Essential and selection Gizmos

6.1.1 Essential gizmos

6.1.2 Selection gizmos

6.1.3 Combo box

6.1.4 Tree view gizmo

6.2 Entry and display Gizmos

6.2.1 Entry gizmos

6.2.2 Bounded and unbounded fields

6.2.3 Validation

6.2.4 Edit fields

6.2.5 Display gizmos

6.2.6 Scroll bars

6.3 New Gizmos

6.3.1 Directly manageable gizmos

6.3.2 Visual gizmos

6.3.3 Adding visual richness to gizmos

Laboratory

There shall be lab exercises cover all the features of visual programming environment.

Reference books

- Alan Cooper, "*The Essential of User Interface Design*", Wiley DreamTech India P. Ltd.
- Evangelos Petroutsos, "*Mastering Visual Basic 6*", BPB Publication

Project - II

BIT278CO

Year: II **Semester: III**

Teaching Schedule Hours/Week			Examination Scheme				
Theory	Tutorial	Practical	Internal Assessment		Final		Total
-	-	4	Theory	Practical	Theory	Practical	100
			-	60	-	40	

Course Objective

After finishing this project, students will be able to develop software using visual programming tool/API.

- Project can be initiated by the project teacher or proposal can be invited by the students.
- Groups of students (up to 4) will be assigned a project work related to any visual programming tool.

Course contents

The students should make the project which has practical significance and should spend four hours per week in the laboratory for 15 weeks. Students must develop the assigned software, submit written report and give oral presentation.

Project evaluation criteria

The practical marks allotted for the project should be evaluated based on the following criteria.

- Title presentation – 10 marks
- Mid-term presentation – 15 marks
- Pre-final submission and presentation – 35 marks

Numerical Methods

BIT280CO

Year: II **Semester: III**

Teaching Schedule Hours/Week			Examination Scheme				
Theory	Tutorial	Practical	Internal Assessment		Final		Total
3	1	2	Theory	Practical	Theory	Practical	150
			20	50	80	-	

Course Objective

This subject aims that enabling students to (a) solve nonlinear equation (b) use interpolation (c) fit curves (d) solve linear equations and (e) perform integration and differentiation, using numerical methods through computers.

Course contents

1. Errors in numerical computation **[3 Hrs]**
   - 1.1 Introduction to numerical method
   - 1.2 Introduction to error
   - 1.3 Sources of error
   - 1.4 General errors formula

2. **Solution of nonlinear equations** **[6 Hrs]**
   - 2.1 Introduction
   - 2.2 Bisection method
   - 2.3 Newton Raphson method
   - 2.4 Fixed point iteration method
   - 2.5 Secant method
   - 2.6 Horner's rule

3. **Interpolation** **[10 Hrs]**
   - 3.1 Introduction
   - 3.2 Finite differences
     - 3.2.1 Forward differences
     - 3.2.2 Backward differences
     - 3.2.3 Central differences
     - 3.2.4 Symbolic relations
   - 3.3 Newton's forward and backward formulae
   - 3.4 Central differences interpolation formula
     - 3.4.1 Gauss forward and backward formula

3.4.2 Stirling's, Bessel's and Everett's formulae

3.5 Lagrange interpolation

3.6 Method of least square method (LSM)

3.6.1 LSM for linear equation ($y = a + bx$)

3.6.2 LSM for quadratic equation ($y = a + bx + cx^2$)

3.6.3 LSM for $y = ax^b$

3.6.4 LSM for $y = ae^{bx}$

4. **System of linear equations** **[11 Hrs]**

4.1 Consistency of a linear system of equations

4.2 Solution of linear system – direct method

4.2.1 Gaussian elimination method

4.2.2 Gauss Jordan method

4.2.3 Matrix inversion

4.3 Solution of linear system – indirect method

4.3.1 Gauss Jacobi iteration method

4.3.2 Gauss Seidel iteration method

4.4 Method of factorization, LU decomposition method

4.5 Eigen vectors and Eigen values, power method

5. **Numerical differentiation and integration** **[8 Hrs]**

5.1 Numerical differentiation for 1^{st} and 2^{nd} order differentiation

5.1.1 Forward formula

5.1.2 Backward formula

5.1.3 Central difference formula

5.2 Numerical integration

5.2.1 Trapezoidal rule

5.2.2 Simpson's 1/3 rule and 3/8 rule

5.2.3 Romberg integration

5.2.4 Gaussian integration

6. **Numerical solution of ordinary differential equations** **[7 Hrs]**

6.1 Introduction

6.2 Euler's method and modified Euler's method

6.3 Rungekutta 2^{nd} order and 4^{th} order methods

6.4 Boundary value problem (finite difference method)

Laboratories

There shall be following lab exercise using high level language.

- Bisection method
- Newton Raphson method
- Fixed-point iteration method
- Secant method
- Horner's rule
- Langrange interpolation
- Newton interpolation
- Least square method for linear equations
- Gauss elimination method
- Gauss Seidel iteration method
- Integration (Trapezoidal rule, Simpson's 1/3 rule and 3/8 rule)
- Euler's method
- Rungekutta 4^{th} order methods

Reference books

- S. S. Sastry, "*Introductory Methods of Numerical Analysis*", PHI
- S. Yakowitz & F. Szidarovszky, "*An Introduction to Numerical Computations*".
- Dr. V. N. Vedamurthy, Dr. N. Ch. S. N. Iyengar, "*Numerical Methods*".
- S. S. Sastry, "*Engineering Mathematics Volume-II*", PHI
- E. Balagurusamy, "*Numerical Methods*".