Syllabus of BCA Fourth Semester Course

The course will consists of Five Theory Papers of 80 marks each and one Practical Paper of 100 marks for which there will be University examinations. Other than the internal evaluation for each Theory Paper which will be of 20 marks and will be evaluated on the basis of classroom performance and internal examination.

The students will be required to answer Five Questions out of which one will be objective and compulsory, where the paper consists of more than one group the students, will be required to answer at least one question from each group.

BCA – 401: Operating Systems

Computer System Overview: Basic Elements, Processor Registers, Instruction Execution, Interrupt, The Memory Hierarchy, Cache Memory, I/O Communication Techniques.

Operating System Overview : Operating System Objectives and Functions, The Evolution of Operating Systems, Major Achievement, Developments Leading to Modern Operating Systems, Microsoft Windows Overview, Traditional UNIX Systems, Linux.

Process Description and Control: Process, Process State, Process Description, Process Control, UNIX SVR4 Process Management.

Threads, SMP and Microkernel: Process and Threads, Symmetric Multiprocessing, Microkernel, Windows Threads and SMP Management, Solaris Thread and SMP Management, Linux Process and Thread Management.

Mutual Exclusion and Synchronization: Principle of Concurrency, Mutual Exclusion, Hardware Support, Semaphores, Monitors, Message Passing, Readers/Writers Problem.

Deadlock and Starvation: Principles of Deadlock, Deadlock Preservation, Deadlock Avoidance, Deadlock Detection, an Integrated Deadlock Strategy, Dining Philosophers Problem, UNIX Concurrency Mechanism, Linux Kernel Concurrency Mechanisms, Solaris Thread Synchronization, Windows Concurrency Mechanisms.

Memory Management: Memory Management Requirements, Memory Partitioning, Paging Segmentation.

Virtual Memory : Hardware and Control Structures, Operating System Software, UNIX and Solaris Memory Management, Linux Memory Management, Windows Memory Management.

Uniprocessor Scheduling: Types of Processor Scheduling, Scheduling Algorithms, Traditional UNIX Scheduling.

Multiprocessor and Real-Time Scheduling: Multiprocessor Scheduling, Real-Time Scheduling, Linux Scheduling, UNIX SVR4 Scheduling, Windows Scheduling.

I/O Management and Disk Scheduling : I/O devices, Organization of the I/O Function, Operating System Design Issue, I/O Buffering, Disk Scheduling, RAID, Disk Cache, UNIX SVR4 I/O, Linux I/O, Windows I/O.

File Management: Overview, File Organization and Access, File Directories, File Sharing, Record Blocking, Secondary Storage Management, UNIX File Management, Linux Virtual File System, Windows File System.

Networking: The Need for a Protocol Architecture, the TCP/IP Protocol Architecture, Sockets, Linux Networking.

Distributed Processing, Client/Server and Clusters: Client/Server Computing, Distributed Message Passing, Remote Procedure Calls, Clusters, Windows Cluster Server, Sun Cluster, Beowulf and Linux Clusters.

Distributed Process Management: Process Migration, Distributed Global States, Distributed Mutual Exclusion, Distributed Deadlock.

Security: Security Threats, Protection, Intruders, Malicious Software, Trusted Systems, Windows Security.

Bibliography and References:

1. William Stalling, Operating System Internals and Design Principles, Pearson Education.

BCA – 402: UML and JAVA

Unified Modeling Language: UML, Goals of UML, Use UML, Diagrams of UML – Use Case Diagram, Class Diagram, Sequence, Diagram, Collaboration Diagram, Statechart Diagram, Activity Diagram, Component Diagram, deployment Diagram.

Use Cases: Introduction to Use Case – System, Actor, Use Cases, Use Case Bundles; Documenting Use Cases – Use Case Diagram, Sequence Diagram, Textual Description; Guideline for Developing Use Cases – Avoiding Analysis Paralysis, Identifying Actors, Identifying High-Level and Essential Use Case, Establishing Use Case Bundles, Developing Use Case Details, Identifying Supporting Use Cases, Developing Boundary Use Case; Contracts.

Finding the Objects: Object-Oriented Analysis: Model of an Application Domain, Building the OO Model, Identification of Objects, Current Techniques, Traditional techniques.

Identifying Responsibilities: Object, Attributes – Descriptive Attributes, Naming Attributes; Service, Method, Identifying Attributes, Specifying Attributes, Identifying Services, Specifying Services.

Static Behavior : Behavior, Techniques for Specifying static Behavior, Techniques for Specifying Control, Techniques for Documenting Control, techniques for Documenting Static Behavior.

Dynamic Behavior: Techniques for Identifying Dynamic Behavior, Identifying and Specifying Events, Specifying Dynamic Behavior, Documenting Dynamic Behavior.

Relationships : Relationships, Generalization/Specialization, Identifying Generalization/ Specialization, Object Aggregation, Classification of Aggregation, Links between Objects, Identifying and Specifying Links and Aggregations, Managing Relationships, Documenting Relationships.

Rules: Rules, Identifying Declarative Statements, Specifying and Documenting Rules, Mapping Rules to the Proper OO Concept, Documenting the Rules Using UML, Implementing Rules.

The Model : Concepts and Object-Oriented Model, Documenting Concepts Using UML, Refining the Model, Subsystems, Organizing Subsystems, Identifying Subsystems, Documenting Subsystems.

Design: System Design, Detailed Design.

Implementation: Implementing Static Behavior, Implementing Dynamic Behavior, Instantiating and Deleting Objects, Implementing Generalization/Specialization, Implementing More relationships.

Bibliography and References:

1. *Martin, UML for Java Programmers, Pearson Education.*
2. *Stevens, Using UML 2/e, Pearson Education.*

BCA – 403: Visual Basic Programming

Visual Basic Application: Introduction, Developing a Visual Basic Application, Using the Application Wizard.

The Visual Basic Environment and Help System: The Visual Basic Interface, Using the Menu Bars, Toolbars, Toolbox, The SDI Environment, Custom Controls and Components.

Customizing a Form and Writing Simple Programs: Starting a New Project, The Properties Windows, Common Form Properties, Making a Form Responsive, Printing a Visual Representation of a Form, Types, Creating Standalone Windows Programs.

Steps in Building the User Interface: Creating Controls, Common Controls of Toolbox, Navigating between Controls, Message Boxes, The Grid, The Display in Visual Basic, The ASCII Representation of the Forms.

First Step in Programming: The Code Window, Visual Basic Editing tools, Statements in Visual Basic, Variables, Setting Properties with Code, Data Types, Working with Variables, More on Strings, More on Numbers, Constants, Input Boxes.

Displaying Information: Displaying Information on a Form, The Format Function, Picture Boxes, Rich Text Boxes, The Printer Object.

Controlling Program Flow: Determinate Loops, Indeterminate Loops, Making a Decisions, Select Case, Nested If…Then's, The GoTo Statements.

Built-In Functions: String Functions, The Like Function and Fuzzy Searching, The RND Function, BitTwiddling Functions, Numeric Functions, Date, Time and Financial Functions.

Writing Your Own Functions and Procedures: Function Procedures, Sub Procedures, Advanced Uses of Procedures and Functions, Using the Object Browser to Navigate among Subprograms.

Organizing Information via Code: Lists – One-Dimensional Arrays, Multi-Dimensional Arrays, Using Lists and Arrays with Functions and Procedures, The New Array-Based String, Sorting and Searching, Records (UserDefined Types), The With Statement, Enums.

Organizing Information via Controls: Control Arrays, List and Combo Boxes, The Flex Grid Control.

Building Larger Projects : Projects with Multiple Forms, Code Modules, The DoEvent Function and Sub Main, Accessing Windows Functions, Error trapping.

VB Objects and an Introduction to Object-Oriented Programming : The Object Browser, Manipulating Objects Built into Visual Basic, General Object Variables, Collections, ObjectOriented Programming, Creating an Object in Visual Basic, Building Your Own Classes.

Finishing the Interface : Visual Basic Displays, Other Control on Toolbox, Common Dialog Boxes, The Microsoft Windows Common Controls 6.0, Menus, MDI Forms, Making Form Independent of Resizing and Screen Resolution, Building Help Systems.

Tools and Techniques for Testing, Debugging, and Optimization: Testing, Bugs, The Immediate Windows, The Debugging Tools, Stopping Programs Temporarily, Final Remarks on Debugging.

An Introduction to Graphics : Fundamentals of Graphics, Screen Scales, The Line and Shape Controls, Graphics via Code, Lines and Boxes, Circle, Ellipses, and Pie Charts, Curves.

Monitoring Mouse Activity: The Mouse Event Procedures, Drag & Drop Operations.

Basic File Handling : File Commands, Sequential Files, Random-Access Files, Binary Files, Sharing Files, Adding Licensing Screens, Keeping File Information Secret.

File System Controls and File System Objects: File System Controls, The File System Objects.

Communicating with Other Windows Applications: The Clipboard, The Shell Function, Sending Keystrokes to the Active Windows Application, Overview of COM/OLE, OLE Automation, Building COM/OLE DLL Servers, OLE Drag and Drop.

Recursion: Recursion, Recursive Sorts, Fractals, Uses of Recursion.

Database Development Using Visual Basic: Modern Database, Using the Data Control, Structured Query Language (SQL), An Introduction to Programming with Database Objects, Methods and Events for the Data Control, Monitoring Changes to the Database, The Data Form Wizard.

Building Your Own ActiveX Controls: Testing the Control, Presentation of Your Control, Adding the Functionality, The Life Cycle of a Control, The Full Code for the Numeric Text Box, Sample Code for Using the Control.

Distributing Your Application: Building the Executable, The Package and Development Wizard.

Bibliography and References:

1. Gary Cornell, Visual Basic 6 from the Ground Up, Tata McGraw-Hill.

BCA – 404: C# Programming

Evolution of Object-Oriented Software Development: Problem Solving Using Computers, Styles of Programming, Complexity of Software, Software Crisis, Software Engineering Principles, Natural Way of Solving a Problem, Abstraction, Interface and Implementation, Encapsulation, Comparing Natural Way of Problem Solving with Conventional Programming Method of Problem Solving, Object Model, Classes and Objects, Data Abstraction and Encapsulation, Modularity, Design a Class, Design Strategies in OOP, Comparison of Structured Programming and OOP, Object-Oriented Programming Languages, Requirements of Using OOP Approach, Advantages of Object-Oriented Programming, Limitations of Object-oriented Programming, Features of Object Oriented Programming, Applications of Object-Oriented Programming.

Basic Elements of C# : .NET Framework, C# Language, Features of C#, Compilation of a C# Program, Character Set of C#, Lexical Elements of C#, Escape Sequences, Identifiers, Keywords, Concept of Data, Operators, Punctuators, Preprocessing Directives, Declarations of Primitive Data Types.

C# Program Structure and Simple I/O Operations: Class, C# Program Structure, Method, Instance Methods, A Simple C# Program, Simple I/O Operations, Creation of Objects Using new Operator, Member Access, Invoking a Method, Types of Parameters, Constructor, Destructor, Default Constructor, The this Reference, Access Modifiers, Static Members, Formatted Output, Header of main() Method, Enumeration.

Operators and Expressions: Classification of Operators, Arithmetic Operators, Relational and Equality Operators, Logical Operators, Assignment Operators, Conditional Operator or Ternary Operator, Type Conversion, String Concatenation, The is Operator, The checked and unchecked Operators, Categories of Expressions, Side Effect, Operator Precedence and Associativity.

Statements : Classification of Statements, Expression Statement, Control Flow Statements, Block, Declaration Statement, Empty Statement, Exception Handling, The Lock Statement, The Labeled Statement, The checked and unchecked Statements.

Array and Structures : Arrays in C#, Classification of Arrays, Creation of Arrays, Creation of Regular Arrays, Creation of Jagged Arrays, Rank and Element Type of Arrays, Reading and Writing of Arrays, Creation of TwoDimensional Jagged Arrays, Creation of ThreeDimensional Jagged Arrays, Difference Between Regular and Jagged Arrays, Initialization of Arrays, Initialization of Jagged Arrays, Creation of Field and Local Arrays and Initialization, Features of Arrays, System Array, Passing Array as a Parameter, Application of Arrays, Recursive Methods, Structure, Nested Structures.

Inheritance : Inheritance, Uses of Inheritance, Derived Class, Syntax for Derived Class, Ultimate Base Class, Implementing Inheritance, Types of Inheritance, .NET Building Blocks, Accessibility, namespaces, Inner Classes, Name Hiding Through Inheritance, Virtual and Override Methods, Dynamic Binding, Abstract Method and Abstract Class, Sealed Classes and Sealed Methods, Characteristics of Inheritance, Restrictions on Accessibility, Advantages of Inheritance.

Interface and Operator Overloading: Interface, Declaration and Implementation of Interfaces, Polymorphism in Interfaces, Multilevel Inheritance, Multiple Inheritance, Explicit Interface Member Implementations, Validating Interfaces, Problems in Interfaces Because of Inheritance, Property, Purpose of Property, Classification of Properties, Inheritance and Property, Properties in Interfaces, Indexers, Delegates, Syntax of Delegate, Multicasting Delegates, Publish/Subscribe Design Pattern, The event Feature in C#, Operator Overloading, Operator Method Declaration for Operator Overloading, The User-Defined Type Conversion, Method Overloading, Collection Interfaces, Variable Method Parameter Lists.

Data Structures: Classification of Data Structures, Operations on Data Structures, Commonly Used Data Structures, Implementation of Data Structures.

File Operations and Multithreading: Stream, File Management System, File Operations, File Management Operations, Stream-Oriented Operations, Multitasking, Multithreading, Threads and Operations of Threads, Secondary Threads, Thread Operations, Synchronization.

Bibliography and References:

1. S. Thamarai Selvi & R. Murugesan, A Textbook on C#, Pearson Education.

BCA – 405: Projects

Two Projects based on Theory Paper BCA – 403.

BCA – 406: Projects

Two Projects based on Theory Paper BCA – 404.