

# Nagindas Khandwala College



**Revised Syllabus  
And  
Question Paper Pattern  
Of Course  
Of  
Bachelor of Science Computer Science  
(BSC CS) Programme**

**(Department Of CS) Second Year  
*Semester IV***

**Under Autonomy**

***(To be implemented from Academic Year- 2017-2018)***

**Bachelor of Computer Science (BSC CS) Program**  
*Under Choice Based Credit, Grading and Semester System*  
**Course Structure**

**SYBSC CS**

*(To be implemented from Academic Year- 2017-2018)*

SYBSC CS – SEMESTER IV							
Course Code	Course	Hrs. of Instruction /Week	Exam Duration (Hours)	Maximum Marks			Credits
				CIE	SEE	Total	
1741UCSFA	Core Subject: Fundamentals of Algorithm	3	2 1/2 Hours	25	75	100	2
1742UCSAJ	Core Subject: Advanced Java	3	2 1/2 Hours	25	75	100	2
1743UCSCN	Core Subject: Computer Networks	3	2 1/2 Hours	25	75	100	2
1744UCSSE	Core Subject: Software Engineering	3	2 1/2 Hours	25	75	100	2
1745UCSLA	Core Subject: Linear Algebra Using Python	3	2 1/2 Hours	25	75	100	2
1746UCSNT	Core Subject: .NET Technologies	3	2 ½ Hours	25	75	100	2
1747UCSAD	Skill Enhancement: Android Developer Fundamentals	3	2 ½ Hours	25	75	100	2

1741UCSPR	Core Subject Practical 1: Fundamentals of Algorithm	2	2 Hours		50	50	1
1742UCSPR	Core Subject Practical 2: Advanced Java	2	2 Hours		50	50	1
1743UCSPR	Core Subject Practical 3: Computer Networks	2	2 Hours		50	50	1
1745UCSPR	Core Subject Practical 5: Linear Algebra using Python	2	2 Hours		50	50	1
1746UCSPR	Core Subject Practical 6: .NET technologies	2	2Hours		50	50	1
1747UCSPR	Skill Enhancement: Practical 7: Android Developer Fundamentals	2	2 Hours		50	50	1
	<b>TOTAL</b>	<b>33</b>					<b>20</b>

Course Code	Course	Hrs. of Instruction/week	Exam Duration (Hours)	Maximum Marks			Credits
				CIE	SEE	Total	
1741UCSFA	Core Subject: Fundamentals of Algorithm	3	2 1/2 Hours	25	75	100	2

Sr. No.	Modules / Units
1	<b>UNIT 1</b>
	Introduction to algorithm, Why to analysis algorithm, Running time analysis, How to Compare Algorithms, Rate of Growth, Commonly Used Rates of Growth, Types of Analysis, Asymptotic Notation, Big-O Notation, Omega-Ω Notation, Theta-Θ Notation, Asymptotic Analysis, Properties of Notations, Commonly used Logarithms and Summations, Performance characteristics of algorithms, Master Theorem for Divide and Conquer, Divide and Conquer Master Theorem: Problems & Solutions, Master Theorem for Subtract and Conquer Recurrences, Method of Guessing and Confirming
2	<b>UNIT 2</b>
	Tree algorithms: What is a Tree? Glossary, Binary Trees, Types of Binary Trees, Properties of Binary Trees, Binary Tree Traversals, Generic Trees (N-ary Trees), Threaded Binary Tree Traversals, Expression Trees, Binary Search Trees (BSTs), Balanced Binary Search Trees, AVL (Adelson-Velskii and Landis) Trees Graph Algorithms: Introduction, Glossary, Applications of Graphs, Graph Representation, Graph Traversals, Topological Sort, Shortest Path Algorithms, Minimal Spanning Tree Selection Algorithms: What are Selection Algorithms? Selection by Sorting, Partition-based Selection Algorithm, Linear Selection Algorithm - Median of Medians Algorithm, Finding the K Smallest Elements in Sorted Order
3	<b>UNIT 3</b>
	Algorithms Design Techniques: Introduction, Classification, Classification by Implementation Method, Classification by Design Method Greedy Algorithms: Introduction, Greedy Strategy, Elements of Greedy Algorithms, Advantages and Disadvantages of Greedy Method, Greedy Applications, Understanding Greedy Technique Divide and Conquer Algorithms: Introduction, What is Divide and Conquer Strategy? Divide and Conquer Visualization, Understanding Divide and Conquer, Advantages of Divide and Conquer, Disadvantages of Divide and Conquer, Master Theorem, Divide and Conquer Applications Dynamic Programming: Introduction, What is Dynamic Programming

	Strategy? Properties of Dynamic Programming Strategy, Problems which can be solved using Dynamic Programming, Dynamic Programming Approaches, Examples of Dynamic Programming Algorithms, Understanding Dynamic Programming, Longest Common Subsequence
--	---

Reference Books
Fundamentals of Algorithm
<p><b>Textbook(s):</b></p> <ol style="list-style-type: none"> <li>1. Data Structure and Algorithmic Thinking with Python, Narasimha Karumanchi , CareerMonk Publications, 2016</li> <li>2. Introduction to Algorithm, Thomas H Cormen, PHI</li> </ol> <p><b>Additional References(s):</b></p> <ol style="list-style-type: none"> <li>1. Data Structures and Algorithms in Python, Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, 2016, Wiley</li> <li>2. Fundamentals of Computer Algorithms, Sartaj Sahni and Sanguthevar Rajasekaran Ellis Horowitz, Universities Press</li> </ol>

Practical (1741UCSPR)
<ol style="list-style-type: none"> <li>1. Write Python program to perform matrix multiplication. Discuss the complexity of algorithm used.</li> <li>2. Write Python program to sort n names using Quick sort algorithm. Discuss the complexity of algorithm used.</li> <li>3. Write Python program to sort n numbers using Merge sort algorithm. Discuss the complexity of algorithm used.</li> <li>4. Write Python program for inserting an element into binary tree.</li> <li>5. Write Python program for deleting an element (assuming data is given) from binary tree.</li> <li>6. Write Python program for checking whether a given graph G has simple path from source s to destination d. Assume the graph G is represented using adjacent matrix.</li> <li>7. Write Python program for finding the smallest and largest elements in an array A of size n using Selection algorithm. Discuss Time complexity.</li> <li>8. Write Python program for finding the second largest element in an array A of size n using Tournament Method. Discuss Time complexity.</li> <li>9. Write Python program for implementing Huffman Coding Algorithm. Discuss the complexity of algorithm.</li> <li>10. Write Python program for implementing Strassen's Matrix multiplication using Divide and Conquer method. Discuss the complexity of algorithm.</li> </ol>

Course Code:	Course	Hrs. of Instruction/week	Exam Duration (Hours)	Maximum Marks			Credits
				CIE	SEE	Total	
1742UCSAJ	Core Subject: Advanced Java	3	2 1/2 Hours	25	75	100	2

Sr. No.	Modules / Units
1	<b>UNIT 1</b>
	<p><b>Swing:</b> Need for swing components, Difference between AWT and swing, Components hierarchy, Panes, Swing components: JLabel, JTextField and JPasswordField, JTextAres, JButton, JCheckBox, JRadioButton, JComboBox and JList</p> <p><b>JDBC:</b> Introduction, JDBC Architecture, Types of Drivers, Statement, ResultSet, Read Only ResultSet, Updatable ResultSet, Forward Only ResultSet, Scrollable ResultSet, PreparedStatement, Connection Modes, SavePoint, Batch Updatations, CallableStatement, BLOB &amp; CLOB</p>
2	<b>UNIT 2</b>
	<p><b>Servlets:</b> Introduction, Web application Architecture, Http Protocol &amp; Http Methods, Web Server &amp; Web Container, Servlet Interface, GenericServlet, HttpServlet, Servlet Life Cycle, ServletConfig, ServletContext, Servlet Communication, Session Tracking Mechanisms</p> <p><b>JSP:</b> Introduction, JSP LifeCycle, JSP Implicit Objects &amp; Scopes, JSP Directives, JSP Scripting Elements, JSP Actions: Standard actions and customized actions,</p>
3	<b>UNIT 3</b>
	<p><b>Java Beans:</b> Introduction, JavaBeans Properties, Examples</p> <p><b>Struts 2:</b> Basic MVC Architecture, Struts 2 framework features, Struts 2 MVC pattern, Request life cycle, Examples, Configuration Files, Actions, Interceptors, Results &amp; Result Types, Value Stack/OGNL</p> <p><b>JSON:</b> Overview, Syntax, DataTypes, Objects, Schema, Comparison with XML, JSON with Java</p>

## Reference Books

### Advanced Java

**Textbook(s):**

- 1) Cay S. Horstmann, Gary Cornell, Core Java™ 2: Volume II—Advanced Features Prentice Hall PTR,9th Edition
- 2) Herbert Schildt, Java2: The Complete Reference, Tata McGraw-Hill,5th Edition
- 3) Joe Wigglesworth and Paula McMillan, Java Programming: Advanced Topics, Thomson Course Technology (SPD) ,3rd Edition

**Additional Reference(s):**

- 1) Advanced Java Programming, Uttam K. Roy, Oxford University Press
- 2) *The Java Tutorials*: <http://docs.oracle.com/javase/tutorial/>
- 3) The Java Tutorials of Sun Microsystems Inc

### Practical (1742UCSPR)

1. Develop the presentation layer of Library Management software application with suitable menus.
2. Design suitable database for Library Management System.
3. Develop business logic layer for Library Management System.
4. Develop Java application to store image in a database as well as retrieve image from database.
5. Write a Java application to demonstrate servlet life cycle.
6. Design database for student administration. Develop servlet(s) to perform CRUD operations.
7. Create Employees table in EMP database. Perform select, insert, update, and delete operations on Employee table using JSP.
8. Write a Student class with three properties. The useBean action declares a JavaBean for use in a JSP. Write Java application to access JavaBeans Properties.
9. Design application using Struts2. Application must accept user name and greet user when command button is pressed.
10. Write Java application to encoding and decoding JSON in Java.

Course Code	Course	Hrs. of Instruction /Week	Exam Duration (Hours)	Maximum Marks			Credits
				CIE	SEE	Total	
1743UCSCN	Core Subject Computer Network	3	2 ½ Hours	25	75	100	2

Sr. No.	Modules / Units
1	<p><b>UNIT 1</b></p> <p><b>Introduction Network Models:</b> Introduction to data communication, Components, Data Representation, Data Flow, Networks, Network Criteria, Physical Structures, Network types, Local Area Network, Wide Area Network, Switching, The Internet, Accessing the Internet, standards and administration Internet Standards. Network Models, Protocol layering, Scenarios, Principles of Protocol Layering, Logical Connections, TCP/IP Protocol Suite, Layered Architecture, Layers in the TCP/IP Protocol Suite, Encapsulation and Decapsulation, Addressing, Multiplexing and Demultiplexing. Detailed introduction to Physical Layer, Detailed introduction to Data-Link Layer, Detailed introduction to Network Layer, Detailed introduction to Transport Layer, Detailed introduction to Application Layer. Data and Signals, Analog and Digital Data, Analog and Digital Signals, Sine Wave Phase, Wavelength, Time and Frequency Domains, Composite Signals, Bandwidth, Digital Signal, Bit Rate, Bit Length, Transmission of Digital Signals, Transmission Impairments, Attenuation, Distortion, Noise, Data Rate Limits, Performance, Bandwidth, Throughput, Latency (Delay)</p>
2	<p><b>UNIT 2</b></p> <p><b>Introduction to Physical Layer and Data-Link Layer:</b> Digital Transmission digital-to-digital conversion, Line Coding, Line Coding Schemes, analog-to-digital conversion, Pulse Code Modulation (PCM), Transmission Modes, Parallel Transmission, Serial Transmission. Analog Transmission, digital-to-analog Conversion, Aspects of Digital-to-Analog Conversion, Amplitude Shift Keying, Frequency Shift Keying, Phase Shift Keying, analog-to-analog Conversion, Amplitude Modulation (AM), Frequency Modulation (FM), Phase Modulation (PM), Multiplexing, Frequency-Division Multiplexing, Wavelength-Division Multiplexing, Time-Division Multiplexing. Transmission Media, Guided Media, Twisted-Pair Cable, Coaxial Cable, Fiber-Optic Cable. Switching, Three Methods of Switching , Circuit Switched Networks, Packet Switching, Introduction to Data-Link Layer, Nodes and Links, Services, Two Sub-layers, Three Types of addresses, Address Resolution Protocol (ARP). Error Detection</p>



	and Correction, introduction, Types of Errors, Redundancy, Detection versus Correction,
<b>3</b>	<b>UNIT 3</b>
	<p><b>Network layer, Transport Layer</b>  Media Access Control (MAC), random access, CSMA, CSMA/CD, CSMA/CA, controlled access, Reservation, Polling, Token Passing, channelization, FDMA, TDMA, CDMA.  Connecting Devices and Virtual LANs, connecting devices, Hubs, Link-Layer ,Switches, Routers,</p> <p>Introduction to Network Layer, network layer services, Packetizing, Routing and Forwarding, Other Services, IPv4 addresses, Address Space, Classful Addressing.  Unicast Routing, General Idea, Least-Cost Routing, Routing Algorithms, Distance-Vector Routing, Link-State Routing, Path-Vector Routing, Introduction to Transport Layer, Transport-Layer Services, Connectionless and Connection-Oriented Protocols.  Transport-Layer Protocols, Service, Port Numbers, User Datagram Protocol, User Datagram, UDP Services, UDP Applications, Transmission Control Protocol, TCP Services, TCP Features, Segment.</p>

Reference Books
Computer Networks
<p><b>Textbook(s):</b>  1) Data Communications and Networking, Behrouz A. Forouzan, Fifth Edition, TMH, 2013.  2) Computer Network, Andrew S. Tanenbaum, David J. Wetherall, Fifth Edition, Pearson Education, 2011.</p> <p><b>Additional Reference(s):</b>  1) Computer Network, Bhushan Trivedi, Oxford University Press  2) Data and Computer Communication, William Stallings, PHI</p>

Practical (1743UCSPR)
<ol style="list-style-type: none"> <li>Understanding the working of NIC cards, Ethernet/Fast Ethernet/Gigabit Ethernet.</li> <li>Crimping of Twisted-Pair Cable with RJ45connector for Straight-Through, Cross-Over, Roll-Over.</li> <li>To understand their respective role in networks/internet.</li> <li>Problem solving with IPv4, which will include concept of Classful addressing. (supportive Hint: use Cisco Binary Game)</li> <li>Using, linux-terminal or Windows-cmd, execute following networking commands and note the output: <i>ping</i>, <i>traceroute</i>, <i>netstat</i>, <i>arp</i>, <i>ipconfig</i>.</li> <li>Using <b>Packet Tracer</b>, create a basic network of two computers using appropriate network wire.</li> </ol>

7. Using **Packet Tracer**, connect multiple (min.6) computers using layer 2 switch.
8. Using **Packet Tracer**, connect a network in triangular shape with three layer two switches and every switch will have four computer. Verify their connectivity with each other.
9. Using **Packet Tracer**, create a wireless network of multiple PCs using appropriate access point.
10. Using **Wireshark**, network analyzer, set the filter for ICMP, TCP, HTTP, UDP, FTP and perform respective protocol transactions to show/prove that the network analyzer is working.

Course Code	Course	Hrs. of Instruction/Week	Exam Duration (Hours)	Maximum Marks			Credits
				CIE	SEE	Total	
1744UCSSE	Core Subject Software Engineering	3	2 <sup>1/2</sup> Hours	25	75	100	2

Sr. No.	Modules / Units
1	<p><b>UNIT 1</b></p> <p><b>Introduction:</b> The Nature of Software, Software Engineering, The Software Process, Generic Process Model, The Waterfall Model, Incremental Process Models, Evolutionary Process Models, Concurrent Models, Component-Based Development, The Unified Process Phases, Agile Development- Agility, Agile Process, Extreme Programming</p> <p><b>Requirement Analysis and System Modeling:</b> Requirements Engineering, Eliciting Requirements, SRS Validation, Components of SRS, Characteristics of SRS , Object-oriented design using the UML - Class diagram, Object diagram, Use case diagram, Sequence diagram, Collaboration diagram, State chart diagram, Activity diagram, Component diagram, Deployment diagram</p>
2	<p><b>UNIT 2</b></p> <p><b>System Design:</b> System/Software Design, Architectural Design, Low-Level Design Coupling and Cohesion, Functional-Oriented Versus The Object-Oriented Approach, Design Specifications, Verification for Design, Monitoring and Control for Design</p> <p><b>Software Measurement and Metrics:</b> Product Metrics – Measures, Metrics, and Indicators, Function-Based Metrics, Metrics for Object-Oriented Design, Operation-Oriented Metrics, User Interface Design Metrics, Metrics for Source Code, Halstead Metrics Applied to Testing, Metrics for Maintenance, Cyclomatic Complexity, Software Measurement - Size-Oriented, Function-Oriented Metrics, Metrics for Software Quality</p>

	<b>Software Project Management: Estimation</b> in Project Planning Process – Software Scope And Feasibility, Resource Estimation, Empirical Estimation Models – COCOMO II, Estimation for Agile Development, The Make/Buy Decision, <b>Project Scheduling</b> - Basic Principles, Relationship Between People and Effort, Effort Distribution, Time-Line Charts
<b>3</b>	<b>UNIT 3</b>
	<p><b>Risk Management</b> - Software Risks, Risk Identification, Risk Projection and Risk Refinement, RMMM Plan</p> <p><b>Software Quality Assurance:</b> Elements of SQA, SQA Tasks, Goals, and Metrics, Formal Approaches to SQA, Six Sigma, Software Reliability, The ISO 9000 Quality Standards, Capability Maturity Model</p> <p><b>Software Testing</b> : Verification and Validation, Introduction to Testing, Testing Principles, Testing Objectives, Test Oracles, Levels of Testing, White-Box Testing/Structural Testing, Functional/Black-Box Testing, Test Plan, Test-Case Design</p>

Reference Books	
Software Engineering	
<b>Text book(s):</b>	
1) Software Engineering, A Practitioner’s Approach, Roger S, Pressman.(2014)	
<b>Additional Reference(s):</b>	
1) Software Engineering, Ian Sommerville, Pearson Education	
2) Software Engineering: Principles and Practices”,Deepak Jain,OXFORD University Press,	
3) Fundamentals of Software Engineering, Fourth Edition, Rajib Mall, PHI	
4) Software Engineering: Principles and Practices, Hans Van Vliet, John Wiley & Sons	
5) A Concise Introduction to Software Engineering, Pankaj Jalote, Springer	

Course Code:	Course	Hrs. of Instruction/week	Exam Duration (Hours)	Maximum Marks			Credits
				CIE	SEE	Total	
1745UCSLA	Core Subject Linear Algebra Using Python	3	2 ½ hrs	25	75	100	2

Sr. No.	Modules / Units
1	<b>UNIT 1</b>
	<b>Field:</b> Introduction to complex numbers, numbers in Python , Abstracting over fields, Playing with GF(2), Vector Space: Vectors are functions, Vector addition, Scalar-vector multiplication, Combining vector addition and scalar multiplication, Dictionary-based representations of vectors, Dot-product, Solving a triangular system of linear equations. Linear combination, Span, The geometry of sets of vectors, Vector spaces, Linear systems, homogeneous and otherwise
2	<b>UNIT 2</b>
	<b>Matrix:</b> Matrices as vectors, Transpose, Matrix-vector and vector-matrix multiplication in terms of linear combinations, Matrix-vector multiplication in terms of dot-products, Null space, Computing sparse matrix-vector product, Linear functions, Matrix-matrix multiplication, Inner product and outer product, From function inverse to matrix inverse <b>Basis:</b> Coordinate systems, Two greedy algorithms for finding a set of generators, Minimum Spanning Forest and GF(2), Linear dependence, Basis , Unique representation, Change of basis, first look, Computational problems involving finding a basis Dimension: Dimension and rank, Direct sum, Dimension and linear functions, The annihilator
3	<b>UNIT 3</b>
	<b>Gaussian elimination:</b> Echelon form, Gaussian elimination over GF(2), Solving a matrix-vector equation using Gaussian elimination, Finding a basis for the null space, Factoring integers, <b>Inner Product:</b> The inner product for vectors over the reals, Orthogonality, <b>Orthogonalization:</b> Projection orthogonal to multiple vectors, Projecting orthogonal to mutually orthogonal vectors, Building an orthogonal set of generators, Orthogonal complement, <b>Eigenvector:</b> Modeling discrete dynamic processes, Diagonalization of the Fibonacci matrix, Eigenvalues and eigenvectors, Coordinate representation in

	terms of eigenvectors, The Internet worm, Existence of eigenvalues, Markov chains, Modeling a web surfer: PageRank.
--	---

## Reference Books

### Linear Algebra using Python

**Textbook(s):**

1) Coding the Matrix Linear Algebra through Applications to Computer Science Edition 1, PHILIP N. KLEIN, Newtonian Press (2013)

**Additional References:**

1) Linear Algebra and Probability for Computer Science Applications, Ernest Davis, A K Peters/CRC Press (2012).

2) Linear Algebra and Its Applications, Gilbert Strang, Cengage Learning, 4<sup>th</sup> Edition (2007).

3) Linear Algebra and Its Applications, David C Lay, Pearson Education India; 3<sup>rd</sup> Edition (2002)

## Practical (1745UCSPR)

1. Write a program which demonstrates the following:
  - a. Addition of two complex numbers
  - b. Displaying the conjugate of a complex number
  - c. Plotting a set of complex numbers
  - d. Creating a new plot by rotating the given number by a degree 90, 180, 270 degrees and also by scaling by a number  $a=1/2$ ,  $a=1/3$ ,  $a=2$  etc.
2. Write a program to do the following:
  - a. Enter a vector  $u$  as a  $n$ -list
  - b. Enter another vector  $v$  as a  $n$ -list
  - c. Find the vector  $au+bv$  for different values of  $a$  and  $b$
  - d. Find the dot product of  $u$  and  $v$
3. Write a program to do the following:
  - a. Enter two distinct faces as vectors  $u$  and  $v$ .
  - b. Find a new face as a linear combination of  $u$  and  $v$  i.e.  $au+bv$  for  $a$  and  $b$  in  $R$ .
  - c. Find the average face of the original faces.
4. Write a program to do the following:
  - a. Enter an  $r$  by  $c$  matrix  $M$  ( $r$  and  $c$  being positive integers)
  - b. Display  $M$  in matrix format
  - c. Display the rows and columns of the matrix  $M$
  - d. Find the scalar multiplication of  $M$  for a given scalar.
  - e. Find the transpose of the matrix  $M$ .
5. Write a program to do the following:
  - a. Find the vector-matrix multiplication of a  $r$  by  $c$  matrix  $M$  with an  $c$ -vector  $u$ .
  - b. Find the matrix-matrix product of  $M$  with a  $c$  by  $p$  matrix  $N$ .
6. Write a program to enter a matrix and check if it is invertible. If the inverse exists, find the inverse.
7. Write a program to convert a matrix into its row echelon form.

8. Write a program to do the following:
- Enter a positive number N and find numbers a and b such that  $a^2 - b^2 = N$
  - Find the gcd of two numbers using Euclid's algorithm.
9. Write a program to do the following:
- Enter a vector b and find the projection of b orthogonal to a given vector u.
  - Find the projection of b orthogonal to a set of given vectors
10. Write a program to enter a given matrix and an eigen value of the same. Find its eigen vector.

Course Code:	Course	Hrs. of Instruction/week	Exam Duration (Hours)	Maximum Marks			Credits
				CIE	SEE	Total	
1746UCSNT	Core Subject .Net Technologies	3	2 ½ hrs	25	75	100	2

Sr. No.	Modules / Units
1	<p><b>UNIT 1</b></p> <p><b>The .NET Framework:</b>.NET Languages, Common Language Runtime, .NET Class Library</p> <p><b>C# Language Basics:</b> Comments, Variables and Data Types, Variable Operations, Object-Based Manipulation, Conditional Logic, Loops, Methods, Classes, Value Types and Reference Types, Namespaces and Assemblies, Inheritance, Static Members, Casting Objects, Partial Classes</p> <p><b>ASP.NET:</b> Creating Websites, Anatomy of a Web Form - Page Directive, Doctype, Writing Code - Code-Behind Class, Adding Event Handlers, Anatomy of an ASP.NET Application - ASP.NET File Types, ASP.NET Web Folders,</p> <p><b>HTML Server Controls</b> - View State, HTML Control Classes, HTML Control Events, HtmlControl Base Class, HtmlContainerControl Class, HtmlInputControl Class, Page Class, global.asax File, web.config File</p>
2	<p><b>UNIT 2</b></p> <p><b>Web Controls:</b> Web Control Classes, WebControl Base Class, List Controls, Table Controls, Web Control Events and AutoPostBack, Page Life Cycle</p> <p><b>State Management:</b> ViewState, Cross-Page Posting, Query String, Cookies, Session State, Configuring Session State, Application State</p> <p><b>Validation:</b> Validation Controls, Server-Side Validation, Client-Side Validation, HTML5 Validation, Manual Validation, Validation with Regular Expressions</p> <p><b>Rich Controls:</b> Calendar Control, AdRotator Control, MultiView Control</p>

	<p><b>Themes and Master Pages:</b> How Themes Work, Applying a Simple Theme, Handling Theme Conflicts, Simple Master Page and Content Page, Connecting Master pages and Content Pages, Master Page with Multiple Content Regions, Master Pages and Relative Paths</p> <p><b>Website Navigation:</b> Site Maps, URL Mapping and Routing, SiteMapPath Control, TreeView Control, Menu Control</p>
<b>3</b>	<b>UNIT 3</b>
	<p><b>ADO.NET:</b> Data Provider Model, Direct Data Access - Creating a Connection, Select Command, DataReader, Disconnected Data Access</p> <p><b>Data Binding:</b> Introduction, Single-Value Data Binding, Repeated-Value Data Binding, Data Source Controls – SqlDataSource</p> <p><b>Data Controls:</b> GridView, DetailsView, FormView</p> <p><b>Working with XML:</b> XML Classes – XMLTextWriter, XMLTextReader</p> <p><b>Caching:</b> When to Use Caching, Output Caching, Data Caching</p> <p><b>LINQ:</b> Understanding LINQ, LINQ Basics,</p> <p><b>ASP.NET AJAX:</b> ScriptManager, Partial Refreshes, Progress Notification, Timed Refreshes</p>

Reference Books
.NET Technologies
<p><b>Textbook(s):</b></p> <p>1) Beginning ASP.NET 4.5 in C#, Matthew MacDonald, Apress(2012)</p> <p><b>Additional Reference(s):</b></p> <p>1) The Complete Reference ASP .NET, MacDonald, Tata McGraw Hill</p> <p>2) Beginning ASP.NET 4 in C# and VB Imar Spanajaars, WROX</p>

Practical (1746UCSPR)
<ol style="list-style-type: none"> <li>1. Write C# programs for understanding C# basics involving <ol style="list-style-type: none"> <li>a. Variables and Data Types</li> <li>b. Object-Based Manipulation</li> <li>c. Conditional Logic</li> <li>d. Loops</li> <li>e. Methods</li> </ol> </li> <li>2. Write C# programs for Object oriented concepts of C# such as: <ol style="list-style-type: none"> <li>a. Program using classes</li> <li>b. Constructor and Function Overloading</li> <li>c. Inheritance</li> <li>d. Namespaces</li> </ol> </li> <li>3. Design ASP.NET Pages with <ol style="list-style-type: none"> <li>a. Server controls.</li> <li>b. Web controls and demonstrate the use of AutoPostBack</li> <li>c. Rich Controls (Calendar / Ad Rotator)</li> </ol> </li> <li>4. Design ASP.NET Pages for State Management using <ol style="list-style-type: none"> <li>a. Cookies</li> <li>b. Session State</li> <li>c. Application State</li> </ol> </li> <li>5. Perform the following activities <ol style="list-style-type: none"> <li>a. Design ASP.NET page and perform validation using various Validation Controls</li> </ol> </li> </ol>

- b. Design an APS.NET master web page and use it other (at least 2-3) content pages.
- c. Design ASP.NET Pages with various Navigation Controls
- 6. Performing ADO.NET data access in ASP.NET for
  - a. Simple Data Binding
  - b. Repeated Value Data Binding
- 7. Design ASP.NET application for Interacting (Reading / Writing) with XML documents
- 8. Design ASP.NET Pages for Performance improvement using Caching
- 9. Design ASP.NET application to query a Database using LINQ
- 10. Design and use AJAX based ASP.NET pages.

Course Code:	Course	Hrs. of Instruction/ week	Exam Duration (Hours)	Maximum Marks			Credits
				CIE	SEE	Total	
1747UCSAD	Skill Enhancement: Android Developer Fundamentals	3	2 ½ hrs	25	75	100	2

Sr. No.	Modules / Units
1	<b>UNIT 1</b>
	What is Android? Obtaining the required tools, creating first android app, understanding the components of screen, adapting display orientation, action bar, Activities and Intents, Activity Lifecycle and Saving State, Basic Views: TextView, Button, ImageButton, EditText, CheckBox, ToggleButton, RadioButton, and RadioGroup Views, ProgressBar View, AutoCompleteTextView, TimePicker View, DatePicker View, ListView View, Spinner View
2	<b>UNIT 2</b>
	User Input Controls, Menus, Screen Navigation, RecyclerView, Drawables, Themes and Styles, Material design, Providing resources for adaptive layouts, AsyncTask and AsyncTaskLoader, Connecting to the Internet, Broadcast receivers, Services, Notifications, Alarm managers, Transferring data efficiently
3	<b>UNIT 3</b>
	Data - saving, retrieving, and loading: Overview to storing data, Shared preferences, SQLite primer, store data using SQLite database, ContentProviders, loaders to load and display data, Permissions, performance and security, Firebase and AdMob, Publish your app



## Reference Books

### Android Developer Fundamentals

**Textbook(s):**

1) "Beginning Android 4 Application Development", Wei-Meng Lee, March 2012, WROX.

**Additional Reference(s):**

1) <https://developers.google.com/training/courses/android-fundamentals>

2) <https://www.gitbook.com/book/google-developer-training/android-developer-fundamentals-course-practicals/details>

## Practical (1747UCSPR)

1. Install Android Studio and Run Hello World Program.
2. Create an android app with Interactive User Interface using Layouts.
3. Create an android app that demonstrates working with TextView Elements.
4. Create an android app that demonstrates Activity Lifecycle and Instance State.
5. Create an android app that demonstrates the use of Keyboards, Input Controls, Alerts, and Pickers.
6. Create an android app that demonstrates the use of an Options Menu.
7. Create an android app that demonstrate Screen Navigation Using the App Bar and Tabs.
8. Create an android app to Connect to the Internet and use BroadcastReceiver.
9. Create an android app to show Notifications and Alarm manager.
10. Create an android app to save user data in a database and use of different queries.

## Evaluation Scheme

### I. Internal Exam-25 Marks

#### (i) Test– 20 Marks - Duration 40 mins

It will be conducted either as a written test or using any open source learning management system such as Moodle (Modular object-oriented dynamic learning environment)Or a test based on an equivalent online course on the contents of the concerned course(subject)offered by or build using MOOC (Massive Open Online Course)platform.

#### (ii) 5 Marks - Active participation in routine class instructional deliveries

Overall conduct as a responsible student, manners, skill in articulation, leadership qualities demonstrated through organizing co-curricular activities, etc.

**II. External Examination- 75 Marks**

(i) Duration - 2.5 Hours.

(ii) Theory question paper pattern:-

All questions are compulsory		
Question	Based on	Marks
Q.1	Unit I	20
Q.2	Unit II	20
Q.3	Unit III	20
Q.4	Unit I,II and III	15

- All questions shall be compulsory with internal choice within the questions.
- Each Question may be sub-divided into sub questions as a, b, c, d & e, etc & the allocation of Marks depends on the weightage of the topic.

**III. Practical Examination – 50 marks (Duration: 2 Hours)**

- Each practical course carries 50 Marks : 40 marks + 05 marks (journal)+ 05 marks(viva)
- Minimum 75% practical from each core/allied course are required to be completed and written in the journal.

**(Certified Journal is compulsory for appearing at the time of Practical Exam)**