

Anekant Education Society's  
**Tuljaram Chaturchand College of Arts, Science and Commerce, Baramati**  
**Autonomous**

**M.Sc.(Computer Science) Academic Year 2019-2020**

**M.Sc. (Computer Science) I - Credit Structure**

Subject	Semester I	Semester II	Total
Paper – I	4	4	8
Paper – II	4	4	8
Paper – III	4	4	8
Paper - IV	4	4	8
Paper – V	4	4	8
Practical	4	4	8
Practical (Project)	---	4	4
Intro. to Cyber Security – I & II	2	2	4
Human Rights	2	--	2
Certificate Course- I	--	2	2
<b>Total =====</b>	<b>28</b>	<b>32</b>	<b>60</b>

**M.Sc. (Computer Science) II - Credit Structure**

Subject	Semester III	Semester IV	Total
Paper – I	4	<b>Industrial Training Project / Internship (IT) 16</b>	4
Paper – II	4		4
Paper – III	4		4
Paper - IV	4		8
Paper – V	4		8
Practical / Paper VI (Sem IV)	4		8
Practical (Project)	4		8
Certificate Course- II	2	-----	2
Skill Development I & II	2	2	4
			2
<b>Total =====</b>	<b>32</b>	<b>18</b>	<b>50</b>

**Extra Credits:**

1	Human Rights	2 Credits
2	Cyber Security Module I & II	4 Credits
3	Certificate Courses I & II	4 Credits
4	Skill Development I & II	4 Credits
Total Extra Credits =		14 Credits

**Total Credits: Academic Credits (24+28+28+16 = 96) + Extra Credits (14) = 110**

### Paper wise Course Structure For M.Sc. (Computer Science)

No	Class	Sem	Code	Paper	Paper Title	Credit	Exam	Marks
1	M.Sc. - I	I	COMP4101	Theory	Principles of Programming Languages (C)	4	I / E	60 + 40
2			COMP4102	Theory	Cryptography and Network Security (C)	4	I / E	60 + 40
3			COMP4103	Theory	Database Technologies (C)	4	I / E	60 + 40
4			COMP4104	Theory	Design and Analysis of Algorithms (C)	4	I / E	60 + 40
5			COMP4105	Theory	Programming with DOT NET (C)	4	I / E	60 + 40
6			COMP4106	Pract.	Lab Course on DOT NET, PPL & Database Technologies (C)	4	I / E	60 + 40
7			HR-101	----	Human Rights – I	2	----	----
8			CYS-101	----	Introduction to Cyber Security – I	2	----	----

**Note: Credit: 24. Core subjects is compulsory and Extra credits (2+2=4) is also compulsory.**

9	M.Sc. - I	II	COMP4201	Theory	Digital Image Processing (C)	4	I / E	60 + 40
10			COMP4202	Theory	Data Mining and Data Warehousing (C)	4	I / E	60 + 40
11			COMP4203	Theory	Python Programming (C)	4	I / E	60 + 40
12			COMP4204	Theory	<b>Advanced Operating System (EI)</b>	4	I / E	60 + 40
13			COMP4205	Pract.	Lab Course on Python Programming and Advance Operating System (C)	4	I / E	60 + 40
14			COMP4206	Pract.	<b>Project (EII)</b>	4	I / E	60 + 40
15			COMP4207	Theory	<b>Artificial Intelligence (EIII)</b>	4	I / E	60 + 40
16			CC-12	----	Certificate Course – I	2	----	---
17			CYS-102	----	Introduction to Cyber Security – II	2	----	----

**Note: : Credit: 28. Core subjects is compulsory and Extra credits (4) is also compulsory.**

18	M.Sc. - II	III	COMP5301	Theory	Mobile Technologies (C)	4	I / E	60 + 40
19			COMP5302	Theory	Soft Computing (C)	4	I / E	60 + 40
20			COMP5303	Theory	Web Services (C)	4	I / E	60 + 40
21			COMP5304	Theory	<b>Software Architecture &amp; Design Pattern (EI)</b>	4	I / E	60 + 40
22			COMP5305	Pract.	Lab Course-on Mobile Technologies and Web Services (C)	4	I / E	60 + 40
23			COMP5306	Pract.	<b>Project (EII)</b>	4	I / E	60 + 40
24			COMP5307	Theory	<b>Recent Trends in IT (Internet of Things) (EIII)</b>	4	I / E	60 + 40
25			CC-23	----	Certificate Course – II	2	----	----
26			SD-23	----	Skill Development – I	2	----	----

**Note: Credit: 28. Core subjects is compulsory and Extra credits (2+2) is also compulsory.**

27	M.Sc. - II	IV	COMP5401	Project	Industrial Training/ Institutional Project <b>(IT) (Core)</b>	16	I / E	60 + 40
28			SD-23	----	Skill Development – II	2	----	----

**Note: Credit: 16. Core subject is compulsory,**

**Total Credits : Academic Credits(24+28+28+16 = 96) + Extra Credits (14) = 110**

**M.Sc.(Computer Science)- I**

**Semester-I**

**Syllabus**

**A.Y. 2019-20**

**SYLLABUS (CBCS) FOR M.Sc. (Computer Science)-I Sem- I**  
**(w.e.f from june 2019) Academic Year 2019-2020**

Class: M.Sc. (Computer Science)(Semester-I)

Paper Code: **COMP4101**

Title of paper: Principles of Programming Language

Paper: **I**

Credit -4

No. of Lectures 48

**Prerequisites:**

It is assumed that student learning this course have the following background:

- Experience with an OOP language (such as Java or C++)
- Experience with a procedural language (such as C)
- Working knowledge of C, C++, and Java programming.
- Basic algorithms and data structure concepts.

**Objectives:**

- This course will prepare you to think about programming languages analytically:
  - Separate syntax from semantics
  - Compare programming language designs
  - Learn new languages more quickly
  - Use standard vocabulary when discussing languages
  - Understand basic language implementation techniques
- **This course focuses on both:**
  - Theory is covered by the textbook readings, lectures, and on the tests
  - Implementation is covered by the homework assignments

Unit	Title and Contents	No. of lectures
<b>Unit -I</b>	<b>Introduction</b> <ul style="list-style-type: none"> <li>✓ The Art of Language Design</li> <li>✓ The Programming Language Spectrum</li> <li>✓ Why Study Programming Languages?</li> <li>✓ Compilation and Interpretation</li> <li>✓ Programming Environments</li> </ul>	2
<b>Unit-II</b>	<b>Non-Imperative Programming</b> <b>Models: Functional, Logic Languages</b> Common LISP <ul style="list-style-type: none"> <li>✓ Basic LISP Primitives ( FIRST,REST,SETF,CONS,APPEND,</li> <li>✓ LIST,NTHCDR,BUTLAST,LAST,LENGTH,REVERSE,ASSC)</li> <li>✓ Procedure definition and binding, DEFUN, LET</li> <li>✓ Predicates and Conditional,</li> </ul> EQUAL, EQ, EQL, =, MEMBER, LISTP, ATOM,NUMBERP, SYMBOLP, NIL, NULL, IF, WHEN,UNLESS,COND, CASE <ul style="list-style-type: none"> <li>✓ Procedure Abstraction and Recursion Turbo Prolog</li> <li>✓ Introduction, facts, Objects and Predicates, Variables,</li> <li>✓ Using Rules, Controlling execution fail and cut predicates.</li> </ul>	<b>10</b>
<b>Unit-III</b>	<b>Names, Scopes, and Bindings</b> <ul style="list-style-type: none"> <li>✓ The Notion of Binding Time</li> <li>✓ Object Lifetime and Storage Management: Static Allocation, Stack-Based Allocation, Heap-Based Allocation, Garbage Collection</li> </ul>	5

	<ul style="list-style-type: none"> <li>✓ Scope Rules</li> <li>✓ Static Scoping, Nested Subroutines, Declaration Order, Dynamic Scoping</li> <li>✓ The meaning of Names in a Scope-Aliases, Overloading, Polymorphism and Related Concepts</li> <li>✓ The Binding of Referencing Environments-Subroutine Closures, First-Class Values and Unlimited Extent, Object Closures</li> <li>✓ Macro Expansion</li> </ul>	
<b>Unit-IV</b>	<p><b>Data Types</b></p> <ul style="list-style-type: none"> <li>✓ Introduction</li> <li>✓ Primitive Data Types-Numeric Types, Integer, Floating point, Complex, Decimal, Boolean Types, Character Types, Character String Types-Design Issues, Strings and Their Operations, String Length Operations, Evaluation, Implementation of Character String Types.</li> <li>✓ User defined Ordinal types-Enumeration types, Designs, Evaluation, Subrange types, Ada's design, Evaluation, Implementation of user defined ordinal types</li> <li>✓ Array types-Array initialization, Array operations, Rectangular and Jagged arrays, Slices, Evaluation, Implementation of Array Types</li> <li>✓ Associative Arrays-Structure and operations, Implementing associative arrays</li> <li>✓ Record type-Definitions of records, References to record fields, Operations on records, Evaluation, Implementation of Record types</li> <li>✓ Union Types-Design issues, Discriminated versus Free unions, Evaluation, Implementation of Union types</li> <li>✓ Pointer and Reference Types-Design issues, Pointer operations, Pointer problems-Dangling pointers, Lost heap dynamic variables, Pointers in C and C++, Reference types, Evaluation, Implementation of pointer and reference types, Representation of pointers and references, Solution to dangling pointer problem, Heap management</li> </ul>	<b>8</b>
<b>Unit - V</b>	<p><b>Control Flow</b></p> <ul style="list-style-type: none"> <li>✓ Expression Evaluation-Precedence and Associativity, Assignments, Initialization, Ordering Within Expressions, Short-Circuit Evaluation</li> <li>✓ Structured and Unstructured Flow-Structured Alternatives to goto Sequencing</li> <li>✓ Selection-Short-Circuited Conditions, Case/Switch Statements</li> <li>✓ Iteration-Enumeration-Controlled Loops, Combination, Loops, Iterators, Logically Controlled Loops</li> <li>✓ Recursion-Iteration and Recursion, Applicative and Normal - Order Evaluation</li> </ul>	<b>5</b>
<b>Unit -VI</b>	<p><b>Subroutines and Control Abstraction</b></p> <ul style="list-style-type: none"> <li>✓ Fundamentals of Subprograms</li> <li>✓ Design Issues for subprograms</li> <li>✓ Local Referencing Environments</li> <li>✓ Parameter-Passing Methods</li> <li>✓ Parameters That are Subprograms</li> <li>✓ Overloaded Subprograms</li> <li>✓ Generic Subroutines-Generic Functions in C++, Generic Methods</li> </ul>	<b>5</b>

	<ul style="list-style-type: none"> <li>in Java</li> <li>✓ Design Issues for Functions</li> <li>✓ User-Defined Overloaded Operators</li> <li>✓ Coroutines</li> <li>✓ The General Semantics of Calls and Returns</li> <li>✓ Implementing “Simple” Subprograms</li> <li>✓ Implementing Subprograms with Stack-Dynamic Local Variables</li> <li>✓ Nested Subprograms</li> <li>✓ Blocks</li> <li>✓ Implementing Dynamic Scoping</li> </ul>	
<b>Unit-VII</b>	<b>Data Abstraction and Object Orientation</b> <ul style="list-style-type: none"> <li>✓ Object-Oriented Programming</li> <li>✓ Encapsulation and Inheritance-Modules, Classes, Nesting (Inner Classes), Type, Extensions, Extending without Inheritance</li> <li>✓ Initialization and Finalization-Choosing a Constructor, References and Values, Execution Order, Garbage Collection</li> <li>✓ Dynamic Method Binding-Virtual- and Non-Virtual Methods, Abstract, Classes, Member Lookup, Polymorphism, Object Closures</li> <li>✓ Multiple Inheritance-Semantic Ambiguities, Replicated Inheritance, Shared Inheritance, Mix-In Inheritance</li> </ul>	<b>8</b>
<b>Unit-VIII</b>	<b>Concurrency</b> <ul style="list-style-type: none"> <li>✓ Introduction-Multiprocessor Architecture, Categories of concurrency, Motivations for studying concurrency</li> <li>✓ Introduction to Subprogram-level concurrency-Fundamental concepts, Language Design for concurrency, Design Issues</li> <li>✓ Semaphores-Introduction, Cooperation synchronization, Competition Synchronization, Evaluation</li> <li>✓ Monitors-Introduction, Cooperation synchronization, Competition Synchronization, Evaluation</li> <li>✓ Message Passing-Introduction, The concept of Synchronous Message Passing</li> <li>✓ Java Threads-The Thread class, Priorities, Competition Synchronization, Cooperation Synchronization</li> </ul>	<b>5</b>
<b>References:</b> <ol style="list-style-type: none"> <li>1. Scott, Programming Language Pragmatics, 3e(With CD) ISBN 9788131222560 Kaufmann Publishers, An Imprint of Elsevier, USA</li> <li>2. Robert W. Sebesta, Concepts of Programming Languages, Eighth Edition, Pearson Education</li> <li>3. Carl Townsend, Introduction to Turbo Prolog</li> <li>4. Patrick Henry Winston &amp; Berthold Klaus Paul Horn ,LISP 3rd edition –BPB</li> <li>5. M. Gabbrielli, S. Martini, , Programming Languages: Principles and Paradigms, Springer ISBN: 9781848829138</li> </ol>		

Class: M.Sc. (Computer Science)(Semester-I)  
 Title of paper: Cryptography and Network Security  
 Credit -4

Paper Code: COMP4102  
 Paper: II  
 No. of Lectures: 50

**Learning Objectives:**

- To enable students to get sound understanding of Info-Sys-Security, Network Security, Cryptography.
- To equip with knowledge and skills necessary to support for their career in Network Security.
- To develop attitude and interest along with necessary knowledge and skills among the students to encourage them to do further academic studies / research in this area, after the completion of their PG Course.

**Learning Outcomes:** Learn the security concepts and techniques.

Units	Title & Contents	No. of Lectures
Unit – I	<b>Introduction to Security:</b> The Need for Security, Security Approaches, Principles of Security, Types of Attacks	02
Unit – II	<b>Cryptography and techniques:</b> Introduction, Plain Text and Cipher Text, Substitution Techniques, Transposition Techniques, Encryption and Decryption, Symmetric and Asymmetric key cryptography, steganography.	06
Unit – III	<b>Symmetric Key Algorithms and AES:</b> Algorithm Types and Modes, Overview of Symmetric Key Cryptography, DES, IDEA, Blowfish	06
Unit – IV	<b>Asymmetric Key Algorithms, Digital Signature and RSA:</b> Brief History of Asymmetric Key Cryptography, overview, RSA Algorithm, Comparison between Symmetric & Asymmetric Key Algorithms, Digital Signature	06
Unit – V	<b>Digital Certificates and Public Key Infrastructure (PKI):</b> Introduction, Digital Certificates, private key management,	03
Unit – VI	<b>Internet Security Protocols:</b> Basic Concepts, SSL, TLS, SHTTP, TSP, SET, SSL v/s SET, 3-D Secure Protocol, Electronic Money, Email Security, WAP Security, Users Authentication and Kerberos: Authentication Basics, Password, Authentication Tokens, Certificate based Authentication, Biometric Authentication, Kerberos, Security Handshake pitfalls, SSO Approaches.	14
Unit – VII	<b>Network Security, Firewalls and Virtual Private Network (VPN):</b> Brief introduction to TCP/IP, Firewalls, IP Security, VPN, Intrusion	05
Unit – VIII	<b>Case studies on Cryptography and Security:</b> Introduction, Cryptographic solutions, Secure inter branch payment transaction, Denial of services(DOS) attacks, IP Spooling Attack, Cross site scripting vulnerability(CSSV), Contract Signing, Secret Splitting, Virtual Electronics, Cookies and Privacy.	08
<b>Reference:</b> <ul style="list-style-type: none"> <li>➤ Cryptography and Network Security : Atul Kahate</li> <li>➤ Computer Network Security : Kizza, Springer</li> <li>➤ Network Security , Harrington, Elsevier</li> </ul>		

Class: M.Sc. (Computer Science)(Semester-I)

Paper Code: COMP4103

Title of paper: Database Technologies

Paper: III

Credit -4

No. of Lectures 52

**Prerequisites:** Knowledge of RDBMS

**Course objectives :**

1. To study the usage and applications of Parallel and Distributed databases, Object relational database and emerging systems.
2. To acquire knowledge on NoSQL databases.

**Course outcomes :**

1. Compare different database technologies.
2. Compare and contrast NoSQL databases with RDBMS.

<b>Units</b>	<b>Title &amp; Contents</b>	<b>No. of Lectures</b>
Unit – I	<b>Parallel and Distributed Databases</b> 1.1 Database System Architectures: Centralized and Client-Server Architectures, Server System Architectures, Parallel Systems, Distributed Systems 1.2 Parallel Databases : I/O Parallelism, Inter and Intra Query Parallelism, Inter and Intra operation Parallelism 1.3 Distributed Database Concepts: Distributed Data Storage, Distributed Transactions, Commit Protocols, Concurrency Control, Distributed Query Processing, Three Tier Client Server Architecture, Case Studies	10
Unit – II	<b>Object and Object Relational Databases</b> 2.1 Concepts for Object Databases: Object Identity, Object Structure, Type Constructors, Encapsulation of Operators, Methods, Persistence, Type and Class Hierarchies, Inheritance, Complex Objects, Object Database Standards 2.2 Languages and Design : ODMG Model, ODL, OQL 2.3 Object Relational and Extended Relational Systems: Object Relational features in SQL/Oracle 2.4 Case Studies	10
Unit – III	<b>XML Databases</b> 3.1 XML Data Model 3.2 DTD 3.3 XML Schema 3.4 XML Querying 3.5 Web Databases 3.6 Information Retrieval 3.7 Data Warehousing 3.8 Data Mining	06
Unit – IV	<b>Mobile Databases</b> 4.1 Location and Handoff Management 4.2 Effect on Mobility on Data Management	10

	4.3 Location Dependent Data Distribution 4.4 Mobile Transaction Models 4.5 Concurrency Control 4.6 Transaction Commit Protocols 4.7 Mobile Database Recovery Schemes 4.8 Examples: Oracle Database Lite, Microsoft SQL Server Compact	
Unit – V	<b>Introduction to NoSQL</b> 5.1 Concepts and evolution 5.2 History of NoSQL 5.3 Different NoSQL products : MongoDB, Couch DB, Cassandra 5.4 Exploring MongoDB 5.5 Advantages of MongoDB over RDBMS 5.6 Interfacing and Interacting with NoSQL 5.7 Sharding 5.8 Replication	08
Unit – VI	<b>Working with NoSQL</b> 6.1 NoSQL Storage Architecture 6.2 CRUD operations with MongoDB 6.3 Querying, Modifying and Managing NoSQL data stores 6.4 Indexing and ordering datasets 6.5 Surveying database internals 6.6 Migrating from RDBMS to NoSQL 6.7 Implementing NoSQL with PHP	08

**References:**

1. Henry Korth, Abraham Silberschatz and S. Sudarshan, “Database System Concepts”, Sixth Edition ,McGraw Hill, 2011.
2. M. Tamer Ozsü and Patrick Valduriez, “Principles of Distributed Database Systems”, Third Edition, Springer, 2011.
3. Thomas Connolly and Carolyn Begg, “Database Systems, A Practical Approach to Design, Implementation and Management”, Fourth Edition, Pearson Education, 2008, Fifth Edition, Pearson Education, 2010, Sixth Edition, Pearson Education, 2015.
4. R. Elmasri, S.B. Navathe, “Fundamentals of Database Systems”, Seventh Edition, Pearson Education, 2017.
5. C.J.Date, A. Kannan, S. Swamynathan, “An Introduction to Database Systems”, Eighth Edition, Pearson Education, 2006.
6. Dan Sullivan, “NoSQL for Mere Mortals”, First Edition, Pearson Education, 2015.
7. Kristina Chodorow, “MongoDB-The Definitive Guide”, Second Edition, O’Reilly, 2013.

Class: M.sc.(Computer science) (Semester-I)  
 Title of Paper: Design & Analysis of Algorithm  
 Credit:4

Paper Code:COMP4104  
 Paper: IV  
 No. of lectures:52

**Learning Objectives:** Student successfully completing this course will be able to

- Understand Basic Algorithm Analysis techniques and the use of asymptotic notation
- Understand different design strategies
- Understand the use of data structures in improving algorithm performance
- Understand classical problem and solutions
- Learn a variety of useful algorithms
- Understand classification of problems

**Learning Outcome**

- Basic algorithms and data structure concepts.
- Basic programming concepts

Units	Title and Contents	No. of Lectures
Unit –I	<b>Design strategies</b> Algorithm definition, space complexity, time complexity, worst case –best case –average case complexity, asymptotic notation, sorting algorithms (insertion sort, heap sort) sorting in linear time, searching algorithms, recursive algorithms ( Tower of Hanoi , Permutations). <b>Divide and conquer</b> control abstraction, binary search, merge sort, Quick sort, Strassen’s matrix Multiplication	12
Unit –II	<b>Advanced Design and Analysis Techniques</b> <b>Greedy method</b> knapsack problem, job sequencing with deadlines, minimum-cost spanning trees, Kruskal and Prim’s algorithm, optimal storage on tapes, optimal merge patterns, Huffman coding <b>-Dynamic programming</b> matrix chain multiplication, single source shortest paths, Dijkstra’s algorithm, Bellman- ford algorithm , all pairs shortest path, longest common subsequence, string editing, 0/1 knapsack problem, Traveling salesperson problem.	16
Unit – III	<b>Decrease and conquer</b> DFS and BFS, Topological sorting, connected components	6
Unit – IV	<b>Backtracking</b> General method, 8 Queen’s problem, Sum of subsets problem, graph coloring problem, Hamiltonian cycle	4
Unit – V	<b>Branch and Bound Technique</b> FIFO, LIFO, LCBB, TSP problem, 0/1 knapsack Problem	4
Unit – VI	<b>Transform and conquer</b> Horner’s Rule and Binary Exponentiation – Problem Reduction	4
Unit – VII	<b>Problem classification</b> Nondeterministic algorithm, The class of P, NP, NP-hard and NP-Complete problems, significance of Cook’s theorem	2

**References:**

1. Ellis Horowitz, Sartaj Sahni & Sanguthevar Rajasekaran, Computer Algorithms, Galgotia.
2. T. Cormen, C. Leiserson, & R. Rivest, Algorithms, MIT Press, 1990
3. A. Aho, J. Hopcroft, & J. Ullman, The Design and Analysis of Computer Algorithms, Addison Wesley, 1974
4. Donald Knuth, The Art of Computer Programming (3 vols., various editions, 1973-81), Addison Wesley
5. Steven Skiena, The Algorithm Manual, Springer ISBN:9788184898651
6. Jungnickel, Graphs, Networks and Algorithms, Springer, ISBN: 3540219056

Class: M.sc. (Computer science) (Semester-I)

Paper Code:COMP4105

Title of Paper: Programming with DOTNET

Paper: V

Credit:4

No.of lectures:50

**Prerequisites –**

- Knowledge of object-oriented programming concepts such as data abstraction, encapsulation, inheritance, and polymorphism.
- Familiarity with programming language such as C++ and/or Java.
- Knowledge of web development

**Learning Objectives:**

Able to understand the DOTNET framework, C# language features and Web development using ASP.NET

**Learning Outcome:**

Ability to write the Visualized programming and design different real life problems.

<b>Part I : C#</b>		
<b>1.</b>	<b>Introduction to DOTNET Framework</b> a. Introduction to DOTNET b. DOT NET class framework c. Common Language Runtime i. Overview ii. Elements of .NET application iii. Memory Management iv. Garbage Collector : Faster Memory allocation, Optimizations <b>d. Common Language Integration</b> i. Common type system ii. Reflection API <b>e. User and Program Interface</b>	<b>2</b>
<b>2.</b>	<b>Introduction to C#</b> <b>a. Language features</b> i. Variables and Expressions, type conversion ii. Flow Control iii. Functions, Delegates iv. Debugging and error handling, exception handling ( System Defined and User Defined) <b>b. Object Oriented Concepts</b> i. Defining classes, class members, Interfaces, properties ii. Access modifiers, Implementation of class, interface and properties iii. Concept of hiding base class methods, Overriding iv. Event Handling <b>c. Collections, Comparisons and Conversions</b> i. Defining and using collections, Indexers, iterators ii. Type comparison, Value Comparison iii. Overloading Conversion operators, as operator <b>d. Generics</b> i. Using generics, ii. Defining Generics, generic Interfaces, Generic methods, Generic Delegate	<b>8</b>
<b>3.</b>	<b>Window Programming</b> <b>a. Window Controls</b> i. Common Controls	<b>6</b>

	<ul style="list-style-type: none"> <li>ii. Container Controls</li> <li>iii. Menus and Toolbars</li> <li>iv. Printing</li> <li>v. Dialogs</li> <li>vi. <b>Data tools</b></li> </ul> <p><b>b. Deploying Window Application</b></p> <ul style="list-style-type: none"> <li>i. Deployment Overview</li> <li>ii. Adding setup project</li> <li>iii. Building the project : Installation</li> </ul>	
<b>4.</b>	<p><b>Data Handling</b></p> <ul style="list-style-type: none"> <li><b>a. File System Data</b></li> <li><b>b. XML Data</b></li> <li><b>c. Databases and ADO.NET</b></li> </ul>	<b>6</b>
<b>5.</b>	<p><b>Reporting Tools</b></p> <ul style="list-style-type: none"> <li><b>a. Data Report</b></li> <li><b>b. Crystal Report</b></li> </ul>	<b>4</b>
<b>6.</b>	<p><b>Dot NET Assemblies</b></p> <ul style="list-style-type: none"> <li>a. Components</li> <li>b. .NET Assembly features</li> <li>c. Structure of Assemblies</li> <li>d. Calling assemblies, private and shared assemblies</li> </ul>	<b>3</b>
<b>Part II : ASP.NET</b>		
<b>1.</b>	<p><b>Introduction to ASP.NET</b></p> <ul style="list-style-type: none"> <li><b>a. History of Web Programming</b></li> <li><b>b. Basic of Web programming</b></li> </ul>	<b>1</b>
<b>2.</b>	<p><b>Server Controls and Variables, control Structures &amp; Functions</b></p> <ul style="list-style-type: none"> <li>a. Forms, webpages, HTML forms, Webforms</li> <li>b. Request &amp; Response in Non-ASP.NET pages</li> <li>c. Using ASP.NET Server Controls</li> <li>d. Datatypes : Numeric, text, arrays, datacollections</li> <li>e. Overview of Control structures</li> </ul>	<b>4</b>
<b>3.</b>	<p><b>Even Driven Programming and PostBack</b></p> <ul style="list-style-type: none"> <li>a. HTML events</li> <li>b. ASP.NET page events</li> <li>c. ASP.NET Web control events</li> <li>d. Event driven programming and postback</li> </ul>	<b>3</b>
<b>4.</b>	<p><b>Reading from Databases</b></p> <ul style="list-style-type: none"> <li>a. Data pages , b. ADO.NET</li> </ul>	<b>3</b>
<b>5.</b>	<p><b>ASP.NET Server Controls</b></p> <ul style="list-style-type: none"> <li>a. ASP.NET Web Controls</li> <li>b. HTML Server Controls</li> <li>c. Web Controls</li> </ul>	<b>4</b>
<b>6.</b>	<p><b>DOTNET assemblies and Custom Controls</b></p> <ul style="list-style-type: none"> <li>a. Introduction to Cookies, Sessions</li> <li>b. Session events</li> <li>c. State management Recommendations</li> </ul>	<b>2</b>
<b>7.</b>	<p><b>Web Services</b></p> <ul style="list-style-type: none"> <li>a. HTTP, XML &amp; Web services</li> <li>b. SOAP</li> <li>c. Building ASP.NET web service</li> <li>d. Consuming a web service</li> </ul>	<b>2</b>

## **Recommended Text and Reference books:**

1. Beginning Visual C#, Wrox Publication
2. Professional Visual C#, Wrox Publication
3. Inside C#, by Tom Archer ISBN: 0735612889 Microsoft Press Â© 2001, 403 pages
4. Beginning ASP.NET 3.5, Wrox Publication
5. Programming ASP.NET 3.5 by Jesse Liberty, Dan Maharry, Dan Hurwitz, O'Reilly
6. Illustrated C# 2008, Solis, Publication APRESS, ISBN 978-81-8128-958-2
7. Professional C# 4.0 and .NET 4 by Christian Nagel, Bill Evjen, Jay Glynn, Karli Watson,
8. Morgan Skinner, WROX
9. Beginning C# Object-Oriented Programming By Dan Clark , Apress
10. ADO.NET Examples and Best Practices for C# Programmers, By Peter D. Blackburn Apress
11. Database Programming with C#, By Carsten Thomsen, Apress

Class: M.sc.(Computer science) (Semester-I)

Paper Code:COMP4106

Title of Paper: Practical On DOT NET, PPL, Database Technologies

Paper: VI (Lab Course)

Credit:4 ( 3 Hr. Practical /week/batch)

No. of Practicals:12

<b>Console Application</b>	
<b>Assignment 1</b>	Parameter Modifiers (ref , out , params)
<b>Assignment 2</b>	Delegate and Events
<b>Assignment 3</b>	Properties and Indexers
<b>Assignment 4</b>	Inheritance and Interface
<b>Assignment 5</b>	Polymorphism (Method Overloading , Operator Overloading and Method Overriding)
<b>Assignment 6</b>	Exception Handling
<b>Assignment 7</b>	Collections
<b>Assignment 8</b>	Generics
<b>Windows Application</b>	
<b>Assignment 1</b>	Use of Basics Form Controls
<b>Assignment 2</b>	Use of List Box
<b>Assignment 3</b>	Event Handling (Calculator)
<b>Assignment 4</b>	Use of Dialogue Boxes
<b>Assignment 5</b>	Simple Database Operations
<b>Assignment 6</b>	Advanced Database Operations
<b>Assignment 7</b>	Simple Crystal Report
<b>Assignment 8</b>	Advanced Crystal Report
<b>ASP.Net Web Applications</b>	
<b>Assignment 1</b>	Use of Web Controls
<b>Assignment 2</b>	Validation Controls
<b>Assignment 3</b>	Use of CSS
<b>Assignment 4</b>	Database Connectivity
<b>Assignment 5</b>	Database Connectivity (Stored Procedure)
<b>Assignment 6</b>	Use of Master Pages
<b>Assignment 7</b>	Use of Master Pages
<b>Assignment 8</b>	Use of State Management (Cookies , Sessions...)
<b>PPL Assignments</b>	
<b>Assignment 1</b>	LISP
<b>Assignment 2</b>	PROLOG
<b>Database Technologies Assignments</b>	
<b>Assignment 1</b>	Creating database, collections, insert, update & delete documents in NoSQL
<b>Assignment 2</b>	Querying documents in NoSQL

**M.Sc.(Computer Science)- I**

**Semester-II**

**Syllabus**

**A.Y. - 2019 -20**

Class: M.Sc.I (Computer Science) Semester-II

Title of Paper : Digital Image Processing

Credit : 04

Paper Code: COMP4201

Paper: I

No.Of Lecture: 48

**Learning Objectives:**

- To understand the relation between human visual system and machine perception and processing of digital images.
- To provide a detailed approach towards image processing applications like enhancement, segmentation, and compression.

**Learning outcome:**

- Review the fundamental concepts of a digital image processing system.
- Analyze images in the frequency domain using various transforms.
- Evaluate the techniques for image enhancement and image restoration.
- Interpret image segmentation and representation techniques

Unit No.	Contents	No. Of Lectures
1	<b>Introduction to DIP</b> <ul style="list-style-type: none"><li>• Introduction to Digital Image Processing</li><li>• The origins of Digital Image Processing</li><li>• Examples of Fields that use Digital Image Processing</li><li>• Gamma-Ray Imaging</li><li>• X-Ray Imaging</li><li>• Imaging in the Ultraviolet Band</li><li>• Imaging in the Visible and Infrared Bands</li><li>• Imaging in the Microwave Band</li><li>• Imaging in the Radio Band</li><li>• Fundamental steps in Digital Image Processing</li><li>• Components of an Image Processing System</li></ul>	03
2.	<b>Digital Image Fundamentals</b> <ul style="list-style-type: none"><li>• Elements of Visual Perception</li><li>• Light and the Electromagnetic Spectrum</li><li>• Image sensing and Acquisition</li><li>• Image Sampling and Quantization</li><li>• Some Basic Relationships between Pixels</li><li>• An Introduction to the Mathematical Tools Used in Digital Image Processing<ul style="list-style-type: none"><li>○ Array versus Matrix Operations</li><li>○ Linear versus Nonlinear Operations</li><li>○ Arithmetic Operations</li><li>○ Set and Logical Operations</li></ul></li></ul>	06
3.	<b>Intensity Transformation and Spatial Filtering</b> <ul style="list-style-type: none"><li>• Background</li><li>• Some Basic Intensity Transformation Functions</li><li>• Histogram Processing<ul style="list-style-type: none"><li>○ Histogram Equalization</li><li>○ Histogram Matching (Specification)</li><li>○ Local Histogram Processing</li></ul></li><li>• Fundamentals of Spatial Filtering</li><li>• Smoothing Spatial Filters</li><li>• Sharpening Spatial Filters</li><li>• Combining Spatial Enhancement Methods</li></ul>	07

4.	<b>Filtering in the Frequency Domain</b> <ul style="list-style-type: none"> <li>• Background</li> <li>• Preliminary Concepts</li> <li>• Sampling and the Fourier Transform of Sampled Functions</li> <li>• The Discrete Fourier Transform (DFT) of One variable</li> <li>• Extension to Functions of Two Variables</li> <li>• Some Properties of the 2-D Discrete Fourier Transform</li> <li>• The Basics of Filtering in the Frequency Domain</li> <li>• Image Smoothing Using Frequency Domain Filters</li> <li>• Image Sharpening Using Frequency Domain Filters</li> <li>• Selective Filtering</li> </ul>	10
5.	<b>Image Restoration and Reconstruction</b> <ul style="list-style-type: none"> <li>• A Model of the Image Degradation / Restoration Process</li> <li>• Noise Models</li> <li>• Restoration in the Presence of Noise Only- Spatial Filtering</li> <li>• Periodic Noise Reduction by Frequency Domain Filtering <ul style="list-style-type: none"> <li>○ Bandreject Filters</li> <li>○ Bandpass Filters</li> <li>○ Notch Filters</li> </ul> </li> <li>• Estimating the Degradation Function</li> <li>• Inverse Filtering</li> <li>• Minimum Mean Square Error(Wiener) Filtering</li> <li>• Geometric Mean Filter</li> </ul>	06
6.	<b>Morphological Image Processing</b> <ul style="list-style-type: none"> <li>• Preliminaries</li> <li>• Erosion and Dilation</li> <li>• Opening and Closing</li> <li>• The Hit-or-Miss Transformation</li> <li>• Some Basic Morphological Algorithms <ul style="list-style-type: none"> <li>○ Boundary Extraction</li> <li>○ Hole Filling</li> <li>○ Extraction of Connected Components</li> <li>○ Convex Hull</li> <li>○ Thinning</li> <li>○ Thickening</li> <li>○ Skeletons</li> <li>○ Pruning</li> <li>○ Morphological Reconstruction</li> </ul> </li> </ul>	05
7.	<b>Image Segmentation</b> <ul style="list-style-type: none"> <li>• Fundamentals</li> <li>• Point, Line, and Edge Detection <ul style="list-style-type: none"> <li>○ Background</li> <li>○ Detection of Isolated Points</li> <li>○ Line Detection</li> <li>○ Edge Models</li> <li>○ Basic Edge Detection</li> </ul> </li> </ul>	07

	<ul style="list-style-type: none"> <li>○ Edge Linking and Boundary Detection</li> <li>● Thresholding <ul style="list-style-type: none"> <li>○ Foundation</li> <li>○ Basic Global Thresholding</li> <li>○ Optimum Global Thresholding Using Otsu's Method</li> <li>○ Using Image Smoothing to Improve Global Thresholding</li> <li>○ Using Edges to Improve Global Thresholding</li> </ul> </li> <li>● Region-Based Segmentation</li> </ul>	
8.	<p><b>Representation and Description</b></p> <ul style="list-style-type: none"> <li>● Representation <ul style="list-style-type: none"> <li>○ Boundary (Border) Following</li> <li>○ Chain Codes</li> <li>○ Polygonal Approximations Using Minimum-Perimeter Polygons</li> <li>○ Other Polygonal Approximation Approaches</li> <li>○ Signatures</li> <li>○ Boundary Segments</li> <li>○ Skeletons</li> </ul> </li> <li>● Boundary Descriptors <ul style="list-style-type: none"> <li>○ Some Simple Descriptors</li> <li>○ Shape Numbers</li> <li>○ Fourier Descriptors</li> </ul> </li> <li>● Regional Descriptors <ul style="list-style-type: none"> <li>○ Some Simple Descriptors</li> <li>○ Topological Descriptors</li> <li>○ Texture</li> </ul> </li> </ul>	04

### **Reference Books:**

1. Sonka, M., Hlavac, V., Boyle, R. [1999]. Image Processing, Analysis and Machine Vision (2nd edition), PWS Publishing, or (3rd edition) Thompson Engineering, 2007
2. Gonzalez, R. C., Woods, R. E., and Eddins, S. L. [2009]. Digital Image Processing Using MATLAB, 2nd ed., Gatesmark Publishing, Knoxville, TN.
3. Anil K. Jain [2001], Fundamentals of digital image processing (2nd Edition), Prentice-Hall, NJ
4. William K. Pratt [2001], Digital Image Processing (3rd Edition), John Wiley & Sons, NY
5. Burger, Wilhelm and Burge, Mark J. [2008]. Digital Image Processing: An Algorithmic Introduction Using Java, Springer
6. Digital Image Analysis (With CD-ROM), Kropatsch, Springer, ISBN 978038795066
7. Digital Image Processing, 6e (With CD), Jähne, Springer, ISBN:978-3-540-24035-8 2

Class : M.Sc. (Computer Science )(Semester – II)  
 Title of Paper : Data Mining and Data Warehousing  
 Credit : 4

Paper Code :COMP4202  
 Paper: II  
 No. of Lectures :55

**Prerequisites :**

- Basic Knowledge of databases handling.

**Learning Objectives :**

- To study different data preprocessing techniques.
- To introduce the core concepts of data warehousing techniques and implementation.
- To introduce the core concepts of data mining techniques and applications.
- To study advanced data mining techniques.
- To use data mining software on various data sets by using proper algorithms.

**Learning Outcomes :**

- Students will understand both the theoretical and practical aspects data mining.
- Understand basic data mining algorithms, methods, and tools
- Understand data mining principles and techniques:
- Understanding the basic concepts of OLAP.
- Understanding the basic concepts of Data Warehouse.

Unit	Title and Contents	No. of Lectures
<b>Unit – I</b>	<b>1. Data Preprocessing</b> 1.1 Introduction 1.2 Data Processing prerequisites 1.3 Data Objects and Attribute Types 1.3.1 Attribute 1.3.2 Nominal Attributes 1.3.3 Binary Attributes 1.3.4 Ordinal Attributes 1.3.5 Numeric Attributes 1.3.6 Discrete Attributes 1.3.7 Continuous Attributes 1.4 Need for Preprocessing 1.5 Major Tasks in Data Preprocessing 1.5.1 Data Cleaning 1.5.2 Data Integration 1.5.3 Data Reduction 1.5.4 Data Transformation 1.5.5 Data Discretization 1.6 Missing Values 1.7 Noisy Data	<b>4</b>
<b>Unit – II</b>	<b>2. Introduction to Data Warehousing</b> 2.1 Introduction 2.2 Data Warehouse: Basic Concepts 2.2.1 Datawarehouse definition 2.2.2 Comparison of OLTP and OLAP 2.2.3 Datamart 2.2.4 Metadata Repository 2.3 Architecture of Data Warehouse 2.4 Data Warehouse Models	<b>7</b>

	<ul style="list-style-type: none"> <li>2.4.1 Enterprise Warehouse</li> <li>2.4.2 Data Mart</li> <li>2.4.3 Virtual Warehouse</li> <li>2.5 Data Cube and OLAP <ul style="list-style-type: none"> <li>2.5.1 Dimension</li> <li>2.5.2 Fact</li> <li>2.5.3 Measures</li> <li>2.5.4 Dimension Table</li> <li>2.5.5 Fact Table</li> <li>2.5.6 Data Cube</li> <li>2.5.7 Cuboid, Apex Cuboid, Base Cuboid</li> <li>2.5.8 OLAP operations</li> </ul> </li> <li>2.6 Dimensional Data Modeling <ul style="list-style-type: none"> <li>2.6.1 Star Schema</li> <li>2.6.2 Snowflake Schema</li> <li>2.6.3 Fact Constellation Schema</li> </ul> </li> </ul>	
<b>Unit – III</b>	<p><b>3. Introduction to Data Mining</b></p> <ul style="list-style-type: none"> <li>3.1 Introduction</li> <li>3.2 Data Mining : Basic Concepts</li> <li>3.3 Knowledge Discovery in Databases Process</li> <li>3.4 Data Mining Tasks <ul style="list-style-type: none"> <li>3.4.1 Descriptive</li> <li>3.4.2 Predictive</li> </ul> </li> <li>3.5 Data Mining Issues</li> <li>3.6 Data Mining Metrics</li> <li>3.7 Social Implications of Data Mining</li> <li>3.8 Applications of Data Mining</li> </ul>	<b>6</b>
<b>Unit – IV</b>	<p><b>4. Data Mining Techniques</b></p> <ul style="list-style-type: none"> <li>4.1 Introduction</li> <li>4.2 Frequent item-sets and association rule mining <ul style="list-style-type: none"> <li>4.2.1 Itemset</li> <li>4.2.2 Frequent Pattern</li> <li>4.2.3 Support</li> <li>4.2.4 Confidence</li> <li>4.2.5 Downward-Closure Property</li> <li>4.2.6 Market Basket Analysis</li> <li>4.2.7 Horizontal Data format</li> <li>4.2.8 Vertical Data format</li> <li>4.2.9 Apriori algorithm</li> </ul> </li> <li>4.3 FP-Tree algorithm</li> <li>4.4 Graph Mining <ul style="list-style-type: none"> <li>4.4.1 Frequent Sub-graph mining</li> <li>4.4.2 Apriori-based Approach</li> <li>4.4.3 Pattern growth Approach</li> </ul> </li> <li>4.6 Tree mining</li> </ul>	<b>8</b>
<b>Unit - V</b>	<p><b>5. Classification &amp; Prediction</b></p> <ul style="list-style-type: none"> <li>5.1 Introduction</li> <li>5.2 Decision Tree Learning <ul style="list-style-type: none"> <li>5.2.1 Construction</li> <li>5.2.2 Basic Decision Tree Algorithm</li> <li>5.2.3 Performance</li> <li>5.2.4 Attribute Selection</li> <li>5.2.5 Issues</li> </ul> </li> </ul>	<b>12</b>

	<p>5.2.6 Classification and Regression Tree(CART)</p> <p>5.3 Bayesian Classification</p> <p>5.3.1 Bays Theorem</p> <p>5.3.2 Navie Baysian Classifier</p> <p>5.3.3 Bayesian Network</p> <p>5.3.4 Inference</p> <p>5.3.5 Parameter Learning</p> <p>5.3.6 Structure Learning</p> <p>5.4 Linear Classification</p> <p>5.4.1 Least Squares</p> <p>5.4.2 Perceptron</p> <p>5.4.3 Support Vector Machine(SVM)</p> <p>5.5 Prediction</p> <p>5.5.1 Linear Regression</p> <p>5.5.2 Nonlinear Regression</p>	
<b>Unit – VI</b>	<p><b>6. Accuracy Measures</b></p> <p>6.1 Introduction</p> <p>6.2 Precision</p> <p>6.3 Recall</p> <p>6.4 F-measure</p> <p>6.5 Confusion Matrix</p> <p>6.6 Cross Validation</p> <p>6.7 Bootstrap</p>	<b>3</b>
<b>Unit – VII</b>	<p><b>7. Clustering</b></p> <p>7.1 Introduction</p> <p>7.2 K-means</p> <p>7.3 Expectation Maximization (EM) algorithm</p> <p>7.4 Hierarchical clustering</p> <p>7.5 Correlation clustering</p>	<b>5</b>
<b>Unit - VIII</b>	<p><b>8. Data Mining Trends and Research Frontiers</b></p> <p>8.1 Introduction</p> <p>8.2 Text mining</p> <p>8.2.1 Text Mining Approaches</p> <p>8.2.2 Text Mining Applications</p> <p>8.3 Web Mining</p> <p>8.3.1 Web Mining Tasks</p> <p>8.3.2 Web Mining Applications</p> <p>8.3.3 Basic introduction of Mining Sequence Data</p> <p>a) Mining of Time-Series Data</p> <p>b) Mining of Symbolic Sequences Data</p> <p>c) Mining of Biological Sequences Data</p> <p>d) Mining of Spatial Data</p> <p>e) Mining of Visual and Audio Data</p>	<b>6</b>
<b>Unit – IX</b>	<p><b>9. Software for data mining</b></p> <p>9.1 Introduction</p> <p>9.2 The Explorer</p> <p>9.3 The Knowledge flow interface</p> <p>9.4 Experimenter</p> <p>9.5 Command Line Interface</p> <p>9.6 Decision Tree with the help of weka</p> <p>9.7 Apriori Algorithm with the help of weka</p>	<b>4</b>

## References :

1. Data Mining: Concepts and Techniques , Jiawei Han, Micheline Kamber, Jian Pei, Elsevier Morgan Kaufmann Publishers.
2. Introduction to data mining : Pang Ning Tan, Michael Steinbach, Vipin Kumar
3. The WEKA Workbench Eibe Frank, Mark A. Hall, and Ian H. Witten Online Appendix for “Data Mining: Practical Machine Learning Tools and Techniques” Morgan Kaufmann, Fourth Edition, 2016
4. [Research-Papers]: Some of the relevant research papers that contain recent results and developments in data mining field

Class : M.Sc. (Computer Science )(Semester – II)

Paper Code :COMP4203

Title of Paper : Python Programming

Paper: III

Credit : 4

No. of Lectures : 50

**Prerequisites:**

- To introduce various concepts of programming to the students using Python.
- Students should be able to apply the problem solving skills using Python

**Learning Objectives:** Student successfully completing this course will be able to understand and gain the knowledge of the subject

Units	Title and Contents	No. of Lectures
<b>Unit -I</b>	<b>Introduction to Python Scripting</b> <ul style="list-style-type: none"><li>• Why Scripting is Useful in Computational Science</li><li>• Classification of Programming Languages</li><li>• Productive Pairs of Programming Languages</li><li>• Gluing Existing Applications</li><li>• Scripting Yields Shorter Code, Efficiency</li><li>• Type-Specification (Declaration) of Variables</li><li>• Flexible Function Interfaces</li><li>• Interactive Computing</li><li>• Creating Code at Run Time</li><li>• Nested Heterogeneous Data Structures</li><li>• GUI Programming</li><li>• Mixed Language Programming</li><li>• When to Choose a Dynamically Typed Language</li><li>• Why Python? Script or Program?</li><li>• Application of Python</li><li>• Concept (immutable)</li></ul>	<b>06</b>
<b>Unit -II</b>	<b>Basic Python</b> <ul style="list-style-type: none"><li>• Python identifiers and reserved words</li><li>• Lines and indentation, multi-line statements</li><li>• Comments</li><li>• Input/output with print and input functions,</li><li>• Command line arguments and processing command line arguments</li><li>• Standard data types - basic, none, Boolean (true &amp; False), numbers</li><li>• Python strings</li><li>• Data type conversion</li><li>• Python basic operators (Arithmetic, comparison, assignment, bitwise logical)</li><li>• Python membership operators (in &amp; not in)</li><li>• Python identity operators (is &amp; is not)</li><li>• Operator precedence</li><li>• Control Statements, Python loops, Iterating by subsequence index, loop control statements (break, continue, pass)</li><li>• Mathematical functions and constants (import math), Random number functions</li></ul>	<b>06</b>

<b>Unit – III</b>	<b>Python strings</b> <ul style="list-style-type: none"> <li>• Concept, escape characters</li> <li>• String special operations</li> <li>• String formatting operator</li> <li>• Single quotes, Double quotes, Triple quotes</li> <li>• Raw String, Unicode strings, Built-in String methods.</li> <li>• Python Lists - concept, creating and accessing elements, updating &amp; deleting lists, basic list operations, reverse</li> <li>• Indexing, slicing and Matrices</li> <li>• built-in List functions</li> <li>• Functional programming tools - filter(), map(), and reduce()</li> <li>• Using Lists as stacks and Queues, List comprehensions</li> </ul>	<b>06</b>
<b>Unit - IV</b>	<b>Python tuples and sets</b> <ul style="list-style-type: none"> <li>• Creating &amp; deleting tuples</li> <li>• Accessing values in a tuple</li> <li>• Updating tuples, delete tuple elements</li> <li>• Basic tuple operations</li> <li>• Indexing, slicing and Matrices, built- in tuple functions.</li> <li>• Sets - Concept, operations.</li> </ul>	<b>06</b>
<b>Unit – V</b>	<b>Python Dictionary</b> <ul style="list-style-type: none"> <li>• Concept (mutable)</li> <li>• Creating and accessing values in a dictionary</li> <li>• Updating dictionary, delete dictionary elements</li> <li>• Properties of dictionary keys</li> <li>• built-in dictionary functions and methods.</li> </ul>	<b>04</b>
<b>Unit – VI</b>	<b>Functions</b> <ul style="list-style-type: none"> <li>• Defining a function (def)</li> <li>• Calling a function</li> <li>• Function arguments - Pass by value, Keyword Arguments, default arguments</li> <li>• Scope of variable - basic rules</li> <li>• Documentation Strings</li> <li>• Variable Number of Arguments</li> <li>• Call by Reference</li> <li>• Order of arguments (positional, extra &amp; keyword)</li> <li>• Anonymous functions</li> <li>• Recursion</li> <li>• Treatment of Input and Output Arguments</li> <li>• Unpacking argument lists</li> <li>• Lambda forms</li> <li>• Function Objects</li> <li>• function ducktyping &amp; polymorphism</li> <li>• Generators (functions and expressions) and iterators, list comprehensions</li> </ul>	<b>08</b>

<b>Unit – VII</b>	<b>Files and Directories</b> <ul style="list-style-type: none"> <li>• Creating files</li> <li>• Operations on files (open, close, read, write)</li> <li>• File object attributes, file positions, Listing Files in a Directory</li> <li>• Testing File Types</li> <li>• Removing Files and Directories</li> <li>• Copying and Renaming Files</li> <li>• Splitting Pathnames</li> <li>• Creating and Moving to Directories</li> <li>• Traversing Directory Trees</li> <li>• Illustrative programs: word count, copy file</li> </ul>	<b>06</b>
<b>Unit – VIII</b>	<b>Python Classes / Objects</b> <ul style="list-style-type: none"> <li>• Object oriented programming and classes in Python - creating classes, instance objects, accessing members</li> <li>• Data hiding (the double underscore prefix)</li> <li>• Built-in class attributes</li> <li>• Garbage collection : the constructor</li> <li>• Overloading methods and operators</li> <li>• Inheritance - implementing a subclass, overriding methods</li> <li>• Recursive calls to methods</li> <li>• Class variables, class methods, and static methods</li> </ul>	<b>08</b>
<b>Unit – IX</b>	<b>Python Exceptions</b> <ul style="list-style-type: none"> <li>• Exception handling : assert statement</li> <li>• Except clause - with no exceptions and multiple exceptions</li> <li>• Try - finally, raising exceptions, user-defined exceptions.</li> </ul>	<b>02</b>

**Reference Books:**

1. Introducing Python- Modern Computing in Simple Packages – Bill Lubanovic, O,,Reilly Publication
2. Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress
3. Practical Programming: An Introduction to Computer Science Using Python 3, Paul Gries, et al., Pragmatic Bookshelf, 2/E 2014
4. Introduction to Computer Science Using Python- Charles Dierbach, Wiley Publication Learning with Python “, Green Tea Press, 2002
5. E-Books : python\_tutorial. pdf, python\_book\_01.pdf
6. Beginning Programming with Python for Dummies Paperback – 2015 by John Paul Mueller
7. A Beginner’s Python Tutorial: [http://en.wikibooks.org/wiki/A\\_Beginner%27s\\_Python\\_Tutorial](http://en.wikibooks.org/wiki/A_Beginner%27s_Python_Tutorial).
4. Introduction to Artificial Intelligence and Expert System, Prentice Hall of India Pvt. Ltd., New Delhi, 1997, 2nd Printing, by Dan Patterson

Class: M.Sc. (Computer Science)(Semester-II)

PAPER CODE: COMP4204

Title of paper: Advanced Operating System

PAPER –IV

Credit -4

No. of Lectures 50

**Prerequisites:**

- Working knowledge of C programming.
- Basic Computer Architecture concepts.
- Basic algorithms and data structure concepts.

**Learning Objectives:**

Students successfully completing this course will be able to:

- Teaches Advanced Operating Systems Concepts using Unix/Linux and Windows as Representative examples.
- Strikes a delicate balance between theory (covered in TextBook-2,3) and practical applications (covered in TextBook-1, 4).
- In fact, most Units start with the theory and then switches focus on how the concepts are implemented in a C program.
- Describes the programming interface to the Unix/Linux system - the system call interface.
- It is intended for anyone writing C programs that run under Unix/Linux.it concludes with an overview of Windows Threads Management.
- Finally it includes with an overview of Android Operating System.

**Learning Outcome:**

- This course provides an understanding of the functions of Operating Systems. It also provides an insight into functional modules of Operating Systems.

Unit	Title and Contents	No. of lectures
Unit-1	<b>Introduction to UNIX/Linux Kernel</b> <ul style="list-style-type: none"><li>• System Structure, User Perspective, Assumptions about Hardware, Architecture of UNIX Operating System,</li><li>• Introduction to kernel, Types of kernel (monolithic, micro)</li><li>• Concepts of Linux Programming- Files and the Filesystem, Processes, Users and Groups, Permissions, Signals, Interprocess Communication.</li></ul>	03
Unit-2	<b>File and Directory I/O</b> <ul style="list-style-type: none"><li>• Buffer headers, structure of the buffer pool, scenarios for retrieval of a buffer, reading and writing disk blocks, inodes, structure of regular file, open, read, write, lseek, close, pipes, dup ,creat, file sharing, atomic operations, dup2, sync, fsync, and fdasync, fcntl, /dev/fd,stat, fstat, lstat, file types, Set-User-ID and Set-Group-ID, file access permissions, ownership of new files and directories, access function, umask function, chmod and fchmod, sticky bit, chown, fchown, and lchown, file size,file truncation, file systems, link, unlink, remove, and rename functions, symbolic links, symlink and readlink functions, file times, utime, mkdir and rmdir, reading directories, chdir, fchdir, and getcwd, device special files.</li><li>• Mapping Files into Memory, Advice for Normal File I/O, I/O Schedulers and I/O Performance, Directories, Copying and Moving files.</li></ul>	13

<b>Unit-3</b>	<b>Process Environment, Process Control and Process Relationships</b> <ul style="list-style-type: none"> <li>• Process states and transitions, layout of system memory, the context of a process, saving the context of a process, sleep, process creation, signals, process termination, awaiting process termination, invoking other programs, the user id of a process, changing the size of the process, The Shell, Process Scheduling</li> <li>• Process termination, environment list, memory layout of a C program, shared libraries, environment variables, setjmp and longjmp, getrlimit and setrlimit, process identifiers, fork, vfork, exit, wait and waitpid, waitid, wait3 and wait4, race conditions, exec, changing user IDs and group IDs, system function, user identification, process times</li> <li>• The Process ID, Running a New Process, Terminating a Process, Waiting for Terminated Child Processes, Users and Groups, Daemons, Process Scheduling, Yielding the Processor, Process Priorities, Processor Affinity</li> </ul>	<b>13</b>
<b>Unit 4</b>	<b>Memory Management</b> <ul style="list-style-type: none"> <li>• The Process Address Space, Allocating Dynamic Memory, Managing Data Segment, Anonymous Memory Mappings, Advanced Memory Allocation, Debugging Memory Allocations, Stack-Based Allocations, Choosing a Memory Allocation Mechanism, Manipulating Memory, Locking Memory, Opportunistic Allocation, Swapping, Demand Paging.</li> <li>• Disk Management- Disk Structure ,Disk Scheduling algorithm, Numerical exercise based on Disk algorithms, Disk management, Swap Space concept and Management, RAID structure, Disk performance issues</li> </ul>	<b>09</b>
<b>Unit 5</b>	<b>Signal Handling</b> <ul style="list-style-type: none"> <li>• Signal concepts, signal function, unreliable signals, interrupted system calls, SIGCLD semantics, reliable-signal technology, kill and raise, alarm and pause, signal sets, sigprocmask, sigpending, sigsetjmp and siglongjmp, sigsuspend, abort, system function revisited, sleep</li> <li>• Basic Signal Management, Sending a Signal, Signal Sets, Blocking Signals, Advanced Signal Management, Sending a Signal with a Payload.</li> </ul>	<b>05</b>
<b>Unit 6</b>	<b>Windows Thread Management</b> <ul style="list-style-type: none"> <li>• Thread Internals <ul style="list-style-type: none"> <li>✓ Data Structures, Kernel Variables, Performance Counters, Relevant Functions, Birth of a Thread Examining Thread Activity : Limitations on Protected Process Threads, Worker Factories (Thread Pools)</li> </ul> </li> <li>• Thread Scheduling <ul style="list-style-type: none"> <li>✓ Overview of Windows Scheduling, Priority Levels, Windows Scheduling APIs, Relevant Tools, Real-Time Priorities, Thread States, Dispatcher Database, Quantum, Scheduling Scenarios, Context Switching,</li> </ul> </li> </ul>	<b>03</b>
<b>Unit 7</b>	<b>Android Operating System</b> <ul style="list-style-type: none"> <li>• Architecture of the Android Operating System:- <ul style="list-style-type: none"> <li>✓ The Android Software Stack, The Linux Kernel – its functions, essential hardware drivers. Libraries - Surface Manager, Media framework, SQLite, WebKit, OpenGL. Android Runtime - Dalvik Virtual Machine, Core Java</li> </ul> </li> </ul>	<b>04</b>

	Libraries. Application Framework - Activity Manager, Content Providers, Telephony Manager, Location Manager, Resource Manager. Android Application – Activities and Activity Lifecycle, applications such as SMS client app, Dialer, Web browser, Contact manager	
--	---	--

**References:**

1. Operating System Concepts, 8th Edition by GREG GAGNE, PETER BAER GALVIN, ABRAHAM SILBERSCHATZ
2. Linux System Programming, O'Reilly, by Robert Love.
3. Windows Internals, Microsoft Press, by Mark E. Russinovich and David A. Soloman.
4. The Design of the UNIX Operating System, PHI, by Maurice J. Bach.
5. Advanced Programming in the UNIX Environment, Addison-Wesley, by Richard Stevens

**Web links:**

Kernel:

<https://github.com/nul1security/Kernel-and-Types-of-kernels/blob/master/Kernel%20and%20Types%20of%20kernels.md>

Android Developers:

<https://www.edgexkits.com/blog/android-operating-system-advantages/>  
<https://developer.android.com/index.html>

Class: M.sc.(Computer science) (Semester-II)

Title of Paper: Practical On Python Programming & AOS

Credit:4 (3 Hr. Practical/week/batch)

Paper Code:COMP4205

Paper: V (Lab Course)

No. of Practicals:12

<b>Python Assignments</b>	
<b>Assignment 1</b>	Basic python programs
<b>Assignment 2</b>	Strings
<b>Assignment 3</b>	Tuples and sets
<b>Assignment 4</b>	Dictionary
<b>Assignment 5</b>	Functions
<b>Assignment 6</b>	Files and Directories
<b>Assignment 7</b>	Classes/objects
<b>Assignment 8</b>	Exception Handling
<b>AOS Assignments</b>	
<b>Assignment 1</b>	Process management
<b>Assignment 2</b>	Memory Management
<b>Assignment 3</b>	Signal Handling
<b>Assignment 4</b>	Disk Scheduling

Class: M.sc.(Computer science) (Semester-II)

Paper Code:COMP4206

Title of Paper: Project

Paper: VI (Lab Course)

Credit:4 (3 Hr. Practical/week/batch)

No. of Practicals:12

### Instructions for Project :

- The Project can be platform, language and technology independent.
- Project will be evaluated by the project guide.
- Assessment will be done weekly in the respective batch.
- Evaluation will be on the basis of weekly progress of project work, progress report, oral, results and documentation and demonstration.
- You should fill your status of project work on the progress report and get the signature of project guide regularly.
- Progress report should sharply focus how much time you have spent on specific task ? You should keep all sign progress report.

Project will not be accepted, if progress report is not submitted and all the responsibilities remain with student.

The format of Progress Report is :

<b>Roll No. &amp; Name of Student:</b>	
<b>Title of the Project:</b>	
<b>Project Guide Name:</b>	

<b>Sr. No.</b>	<b>Date</b>	<b>Details of Project Work</b>	<b>Project Guide Sign (With Date)</b>
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			

Head

Department of Computer Science

Class: M.sc.(Computer science) (Semester-II)

Paper Code:COMP4207

Title of Paper: Artificial Intelligence

Paper:VII

Credit:4

No.of lectures:48

**Learning Objectives:** Student successfully completing this course will be able to

- Understand and gain the knowledge of the subject

**Learning Outcome:** Concepts of Data structures and Design and Analysis of algorithms

Units	Title and Contents	No. of Lectures
Unit –I	<b>Introduction</b> - What is AI -Goals of AI -AI & related fields -AI technique -Introduction to robotics -Robot components (embodiment, sensors, states, action, brains & brawn, autonomy, arms, legs, wheels, tracks) -languages use in robotics. -latest trends (ASIMO,SOPHIA.....)	8
Unit –II	<b>Problem, Problem Spaces &amp; Search</b> -state space search -production system -search & control strategies -problem characteristics -issues in the design of search program.	6
Unit – III	<b>Heuristics Search Techniques</b> -Heuristics search technique -Generate and test algorithm -Hill climbing(Simple hill climbing, Steepest hill climbing, Simulated annealing) -Best First Search(A* algorithm) -Problem Reduction(AND-OR-Graphs, AO* algorithm) -Constraint Satisfaction -Mean-Ends Analysis	8
Unit - IV	<b>Knowledge Representation</b> -Knowledge representation and mapping -Approaches to knowledge representation -Types of knowledge -Propositional Logic -Predicate Logic -CNF -Resolution -Forward & Backward chaining system	8
Unit – V	<b>Slot &amp; Filler Structures</b> -Introduction <ul style="list-style-type: none"><li>✓ Semantic network</li><li>✓ Inference in semantic net</li><li>✓ Partitioned semantic net</li></ul> -Frames -CD(conceptual dependency)	6

	-Script -CYC(CYC Motivation, CYCL)	
<b>Unit – VI</b>	<b>Game Playing</b> -Introduction -Min-Max algorithm -Adding alpha-beta cutoff -Uncertainty Reasoning(Basic probability axioms, Baye’s rule, Certainty theory, Bayesian classification, Dempster-Shafer Theory)	<b>6</b>
<b>Unit – VII</b>	<b>Learning</b> -Introduction -Rote learning -Learning by Taking Advice -Learning in problem solving(Learning by parameter adjustment, Learning by macro operators, Learning by chunking) -Learning from Example-Induction -Winston Learning Program(Version Spaces, Decision trees) -Explanation Based Learning(EBL)(EBL Architecture, EBL System Schematic)	<b>6</b>

**References:**

1. Computational Intelligence, Eberhart, Elsevier, ISBN 9788131217832
2. Artificial Intelligence: A New Synthesis, Nilsson, Elsevier, ISBN 9788181471901
3. Artificial Intelligence, Tata McGraw Hill, 2nd Edition, by Elaine Rich and Kevin Knight
4. Introduction to Artificial Intelligence and Expert System, Prentice Hall of India Pvt. Ltd., New Delhi, 1997, 2nd Printing, by Dan Patterson



# SYLLABUS (CBCS) FOR M. Sc. (Computer Science)-II (Semester- III)

(w.e.f from Academic Year 2020-2021)

**Class: M. Sc. (Computer Science) (Sem-III)**

**Title of Paper: Mobile Technologies (Core)**

**Credit: 4 (4 Lectures/Week)**

**Paper Code: Comp5301**

**Paper: I**

**No. of lectures: 48**

**Pre-requisites:**

- Concepts of Networking, Wireless communication.
- Familiar with object oriented concepts
- Conversant with OS internals

**Course Objectives:**

- Students can understand basic concepts of mobile technology and mobile platforms
- Students can understand concepts related to wireless communication.
- Create native android applications using basic concepts
- Understand UI design or controls available in android.

<b>Title and Contents</b>	<b>No. of Lectures</b>
<b>Unit 1 : Wireless Transmission</b> 1.1 Multiplexing and Modulation 1.2 Spread Spectrum 1.3 Wireless transmission media 1.4 Migration from 2G to 4G 1.5 Wireless LAN advantages 1.6 Wireless LAN architecture 1.7 Mobility in Wireless LAN 1.8 Wireless LAN security	7
<b>Unit 2: Introduction to Mobile Computing</b> 2.1 Introduction and need for Mobile computing 2.2 Mobility and portability 2.3 Mobile and Wireless devices 2.4 Mobile Applications 2.5 Mobile Operating system – IOS, BlackBerry, Windows ,phone, Plam OS, Symbian OS,PhoneGap 2.6 Cellular systems. 2.7 Introduction to IEEE 802.11	6
<b>Unit 3: GSM Technology</b> 3.1 What is GSM? 3.2 GSM Services. 3.3 GSM Network Architecture a. Radio Subsystem b. Network Subsystem c. Radio Subsystem 3.4 Localization and Calling 3.5 Handover 3.6 GSM security 3.7 GPRS 3.8 CDMA in Mobile Communication	8

<b>Unit 4 : Introduction to Android Operating System &amp; Programming</b> 4.1 Features of Android 4.2 Android Architecture 4.3 Fundamental of Android App 4.4 Java for Android 4.5 Activities and Intents 4.6 User Interface 4.7 Services and Broadcast Receivers	8
<b>Unit 5: Android UI Design</b> 5.1 Basic UI Designing (Form widgets ,Text Fields , Layouts ,[dip, dp, sip, sp] versus px) 5.2 Intent (in detail) 5.3 All components (e.g Button , Slider, Image view, Toast) Event Handling 5.4 Adapters and Widgets 5.5 Menu	5
<b>Unit 6:Android Thread and Notification</b> 6.1 Threads running on UI thread (run On Ui Thread) 6.2 Worker thread 6.3 Handlers & Runnable 6.4 AsyncTask (in detail) 6.5 Broadcast Receivers Services and notifications 6.6 Toast 6.7 Alarms	4
<b>Unit 7: iOS Fundamentals</b> 7.1 <b>Introduction</b> - What is IOS ,IOS Architecture, Frameworks, Application Life Cycle, Features 7.2 <b>Swift</b> - Introduction to Swift ,General Concepts of Swift 7.3 <b>Xcode</b> - Introduction to Xcode , Navigator, Editor Utility, Tools, Console, Document, Simulator, Instruments 7.4 <b>Startup</b> - Application Templates, Introduction to Storyboard , Hello World Application, How ‘Hello World’ Working, Debugging Database, Plist, Preference, Sqlite Web Service, Restful Web Service (JSON & XML)	6
<b>Unit 8: Introduction to Flutter Mobile App Development</b> 8.1 Flutter – Introduction 8.2 Features of Flutter 8.3 Basics of creating flutter application in the Android Studio. 8.4 Flutter – Architecture of Flutter Application 8.5 Flutter – Introduction to Dart Programming 8.6 Flutter – Introduction to Gestures. 8.7 Introduction to Beacon Mobile technology	04

## References:

1. Schiller Jochen, (2003). Mobile Communications. UK.: Pearson Education
2. Wei-Meng Lee . Beginning Android Application Development. John Wiley & Sons, 2012
3. Talukder Asoke K. Hasan Ahmed,Roopa R Yavagal. Mobile Computing: Technology, Applications and Service Creation
4. Mark Murphy. Beginning Android 3. APress , ISBN 9788132203568
5. Matthijs Hollemans . IOS Apprentice

## Websites :

<https://www.tutorialspoint.com/>

<https://developer.android.com/>

<https://w3resource.com/>

<https://abhiandroid.com/>

**Class: M. Sc. (Computer Science) (Sem-III)**

**Title of Paper: Soft Computing (Core)**

**Credit: 4 (4 Lectures/Week)**

**Paper Code: Comp5302**

**Paper: II**

**No. of lectures: 48**

**Prerequisites :** Probability, First Order Mathematical Logic, Classical Logic, Linear algebra and calculus

**Objective**

- To understand the concepts of how an intelligent system work and its brief development process.
- Be familiar with design of various neural networks & fuzzy logic & Learn genetic programming.

**Learning Outcome:**

- Describe human intelligence and AI
- Explain how intelligent system works.
- Apply basics of Fuzzy logic and neural networks.
- Understand the ideas of fuzzy sets, fuzzy logic and use of human experience relate with neural networks, generalize appropriate rules for inference systems
- Understand the genetic algorithms and other random search procedures.
- Develop some familiarity with current research problems and research methods in Soft Computing Techniques.

<b>Units &amp; Contents</b>	<b>No. Of Lectures</b>
<p style="text-align: center;"><b>1. Introduction to Fuzzy Logic</b></p> <ul style="list-style-type: none"><li>• The Illusion : Ignoring Uncertainty and accuracy</li><li>• Uncertainty and information</li><li>• Fuzzy set and membership</li><li>• Chance versus Fuzziness</li><li>• Classical Sets, Fuzzy Sets</li><li>• Cartesian Product</li><li>• Crisp Relations</li><li>• Fuzzy relations</li><li>• Tolerance and equivalence Relations, Fuzzy Tolerance and equivalence Relations</li><li>• Value assignments, Other Forms of the Composition Operations</li><li>• Features of the membership Function</li><li>• various forms, Fuzzification, Defuzzification to Crisp sets</li><li>• <math>\lambda</math>-Cuts for fuzzy Relations, Defuzzification to Scalars</li></ul>	12
<p style="text-align: center;"><b>2 : Fuzzy logic</b></p> <ul style="list-style-type: none"><li>• Fuzzy Logic</li><li>• Approximate Reasoning, Others forms of implication operations</li><li>• Natural Language, Linguistic Hedges</li><li>• Fuzzy (Ruled-Based) system</li><li>• Graphical technique of inference</li><li>• Membership value assignment-Intuition, Inference.</li></ul>	04
<p style="text-align: center;"><b>3 : Fuzzy System and Classification</b></p> <ul style="list-style-type: none"><li>• Fuzzy System Simulation-<ul style="list-style-type: none"><li>○ Fuzzy Relation</li><li>○ Equations</li><li>○ Nonlinear Simulation Using Fuzzy Systems</li></ul></li></ul>	10

- Fuzzy Associative Memories
- Fuzzy Classification
  - Classification by Equivalence Relations
  - Cluster Analysis
  - Cluster Validity
- c-Means Clustering
- Hard c-Means
- Fuzzy c-Means
- Classification Metric
- Hardening the Fuzzy c-Partition
- Similarity Relations from Clustering
- Fuzzy Arithmetic and Extension Principle
  - Extension Principle
  - Fuzzy Arithmetic
  - Interval Analysis in Arithmetic
  - Approximate Methods of Extension.

#### 4 : Neural Network

12

- Neural networks:
  - Artificial Neural Network: Definition
  - Advantages of Neural Networks Application
  - Scope of Neural Networks
- Fundamental Concept:
  - Artificial Neural Network
  - Biological Neural Network
- Brain vs. Computer-
  - Comparison Between Biological Neuron and Artificial Neuron (Brain vs. Computer)
- Artificial Neurons
- Neural Networks and Architectures:
  - Neuron Abstraction
  - Neuron Single Functions
  - Mathematical Preliminaries
  - Neural Networks Defined
- Architectures:
  - Feed forward and Feedback,
- Salient Properties of Neural Networks
- Geometry of Binary Threshold Neurons and Their Networks:
  - Pattern Recognition and Data Classification
  - Convex Sets
  - Convex Hulls and Linear Separability
  - Space of Boolean Functions
  - Binary Neurons are Pattern Dichotomizers
  - Non-linearly Separable Problems
  - Capacity of a Simple Threshold Logic Neuron

<ul style="list-style-type: none"> <li>○ Revisiting the XOR Problem</li> <li>○ Multilayer Networks</li> <li>○ How Many Hidden Nodes are Enough?</li> </ul>	
<p style="text-align: center;"><b>5 :Introduction to learning</b></p> <ul style="list-style-type: none"> <li>● Learning and Memory: <ul style="list-style-type: none"> <li>○ An Anecdotal Introduction</li> <li>○ Long Term Memory</li> </ul> </li> <li>● The Behavioural Approach to Learning</li> <li>● The Molecular Problem of Memory</li> <li>● Learning Algorithms</li> <li>● Error Correction and Gradient Descent Rules</li> <li>● Learning Objective for TLNs</li> <li>● Pattern Space and Weight Space</li> <li>● Linear Separability</li> <li>● Hebb Network</li> <li>● Perceptron Network</li> <li>● <math>\alpha</math>- Least Mean Square Learning</li> <li>● MSE Error Surface and Its Geometry</li> <li>● Steepest Descent Search with Exact Gradient Information</li> <li>● <math>\mu</math>-LMS: <ul style="list-style-type: none"> <li>○ Approximate Gradient Descent</li> <li>○ Application of LMS to Noise Cancellation.</li> </ul> </li> </ul>	06
<p style="text-align: center;"><b>6 :Genetic Algorithms</b></p> <ul style="list-style-type: none"> <li>● A Gentle Introduction to Genetic Algorithms: <ul style="list-style-type: none"> <li>○ What are Genetic Algorithms?</li> <li>○ Robustness of Traditional Optimization and Search Methods</li> <li>○ The Goals of Optimization</li> </ul> </li> <li>● How are Genetic Algorithms Different from Traditional Methods?</li> <li>● A simple Genetic Algorithm</li> <li>● Genetic Algorithms at Work—a Simulation by hand</li> <li>● Grist for the Search Mill—Important Similarities</li> <li>● Similarity Templates (Schemata) Learning the Lingo.</li> </ul>	04

### Reference Books

1. Fuzzy Logic With Engineering Applications, 3rd Edition By Timothy Ross , Wiley Publication
2. Neural Networks By Satish Kumar, Tata McGraw Hill
3. Introduction to Soft Computing by Deepa & Shivanandan, Wiley Publication
4. Genetic Algorithms in Search, Optimization and Machine Learning By David E. Goldberg, Pearson Education.

**Class: M. Sc. (Computer Science) (Sem-III)**

**Title of Paper: Web Services (Core)**

**Credit: 4 (4 Lectures/Week)**

**Pre-requisites:**

Strong knowledge about Java programming, Good Understanding of Object Oriented Programming concepts. Must be familiar with XML.

**Paper Code: Comp5303**

**Paper: III**

**No. of lectures: 48**

**Objectives:**

- To Understand Web Services and implementation model for SOA.
- To Understand the SOA, its Principles and benefits.
- To understand cloud computing as a web service.
- To understand XML concepts. To understand paradigms needed for testing Web Services.

**Learning Outcomes:** Student will able to :

- Understand the principles of SOA
- Efficiently use market leading environment tools to create and consume web services.
- Identify and select the appropriate framework components in creation of web service solution.

<b>Units &amp; Contents</b>	<b>No. of Lecture</b>
<b>1. Web Service and SOA fundamental</b>	<b>8</b>
1. Introduction : what are Web Services?, Concept of Saas 2. Web services Vs Web based Implementation 3.Characteristics of Web Services: Types of Web services, Functional and non functional properties , State processing , loose Coupling, Service Granularity, Service Synchronization 4.Service Interface and Implementation 5.The Services oriented Architecture (SOA): Roles of Interaction in SOA, Layer of SOA 6.Quality O f Service (QoS) 7.Web Service Interoperability(WS-I) 8.Web Services Vs Components 9.Restful Services 10.Impact and Shortcomings of Web services: Impact of web services	
<b>2. Web Services Architecture</b>	<b>8</b>
1. Introduction 2.Web Services Architecture and It's Characteristics: Web service characteristics , Web Service Architecture 3 Core building blocks of web services 4 Standards and technologies available for implementing web services: SOAP, WSDL,UDDI 5 Web Services Communication Models: RPC-based Communication Model, Messaging-based communication Model 6 Basic steps of implementing web services 7 Developing web services-Enabled applications: Web Services Implementation Using J2EE Environment, Developing Web Services Using J2EE:AN Example 8 Packaging and Deploying the Service: Creating web Service Clients	
<b>3. SOAP: Simple Object Access Protocol</b>	<b>8</b>
1 Introduction 2 Inter-application communication and wire protocols: SOAP as a wire representation, SOAP as a messaging protocol 3 Structure of a SOAP message: SOAP Envelope, SOAP Header, SOAP Body 4 SOAP communication model: RPC-Style, Document –(Message)Style Web Services 5 Building SOAP Web Services 6 developing SOAP Web Services using Java 7 Error handling in SOAP and Advantages and disadvantages of SOAP.	

<b>4. Describing and Discovering Web Services:</b>	<b>12</b>
1 WSDL in the world of Web Services 2 Web Services life cycle: Anatomy of WSDL definition document, Patterns Of Operations 3 WSDL bindings, WSDL Tools, limitations of WSDL 4 Service discovery: Role of service discovery in a SOA, service discovery mechanisms 5 UDDI – UDDI Registries, uses of UDDI Registry, Searching information in a UDDI Registry, Deleting information in a UDDI Registry, limitations of UDDI.	
<b>5. Cloud Computing:</b>	<b>12</b>
1 Introduction: What is Cloud Computing?, Essential characteristic of cloud, Cloud Deployment Model, History, Benefits, Risk 2 SOA Meets the Cloud : Comparing SOA with Cloud Computing , SOA Deployment :Cloud Computing Vs The ESB 3 Cloud Computing Technologies: Virtualization, SOA, Grid Computing, Utility Computing 4 Cloud Computing Architecture: Front End, Back End 5 Cloud Model : Public Cloud ,Private Cloud, Hybrid Cloud, Benefits, Disadvantages 6 Security and Privacy: Insecure or incomplete data and deletion, Security Planning, Understanding security of cloud ,Security Boundaries, Understanding Data Security, Isolated Access To data 7 Cloud Computing Application: Bossiness, Social, Entertainment	

### **Learning Resources**

#### **Text Book :**

Web Services & SOA Principles and Technology

#### **Reference Book :**

1. Restful Web Services Cookbook By Subbu Allamaraju
2. Service Architecture By Thomas Eri
3. XML, Web Services, and the Data Revolution ,F.P. Coyle, Pearson Education

**Class: M. Sc. (Computer Science) (Sem-III)**

**Paper Code: Comp5304**

**Title of Paper: Software Architecture & Design Pattern Paper: IV (Elective-I)**

**Credit: 4 (4 Lectures/Week)**

**No. of lectures: 48**

**Objectives:**

To Understand and learn the software architecture, its styles, views and pattern for design software with minimum complexity and maintain flexibility

**Prerequisites:** System Analysis and Design, Software Engineering, OOSE, Software project Management, UML

**Outcomes:** Able to assist learner to utilize styles and views to state Architecture, define documentation, analyze the architectural structures and it's Influence on business and development process.

<b>Units &amp; Title and Contains</b>	<b>No of Lect.</b>	<b>Assignments /test</b>
<b>1. Introduction to Software Architecture</b> 1.1 Introduction to concept of Software Architecture 1.2 Definition , Architectural structures 1.3 Need and Influence of software architecture in organization as business and technical aspects 1.4 Architecture Business Cycle 1.5 Introduction – Functional requirements, Technical constraints, Quality Attributes	<b>4</b>	<b>Assignment-1</b>
<b>2. Quality Attribute</b> 2.1 Introduction Quality Attribute 2.2 Documenting Quality Attributes 2.3 Six part scenarios 2.4 Case studies	<b>4</b>	<b>Assignment-2</b>
<b>3. Architectural Views</b> 3.1 Introduction ,Definitions for views 3.2 Structures and views, Representing views, available notations, 3.3 Standard views,4+1 view of RUP, Siemens 4 views SEI's perspectives and views 3.4 Case studies	<b>6</b>	<b>Assignment-3</b> <b>Test-I</b>
<b>4. ARCHITECTURAL STYLES</b> 4.1 Introduction 4.2 Data flow styles with Case study 4.3 Call-return styles with Case study 4.4 Shared Information styles with Case study 4.5 Event styles with Case study	<b>6</b>	<b>Assignment-4</b>
<b>5. Common Software Architectural Patterns</b> 5.1 Layered pattern 5.2 Client-server pattern 5.3 Master-slave pattern 5.4 Pipe-filter pattern, 5.5 Broker pattern 5.6 Peer-to-peer pattern 5.7 Event-bus pattern 5.8 Model-view-controller pattern 5.9 Blackboard pattern 5.10 Interpreter pattern	<b>12</b>	<b>Assignment-5</b> <b>Test-II</b>

<p><b>6. Design Pattern</b></p> <p>6.1 Important Design Patterns</p> <p>6.2 Design Pattern Catalogue, Creational, Structural and behavioral patterns</p> <p>6.3 Structural Decomposition</p> <p>6.4 Organization of Work, Access Control.</p>	<b>10</b>	<b>Assignment-6</b>
<p><b>7. DOCUMENTING THE ARCHITECTURE</b></p> <p>7.1 Good practices, Documenting the Views using UML,</p> <p>7.2 Merits and Demerits of using visual languages</p> <p>7.3 Need for formal languages</p> <p>7.4 Architectural Description Languages ACME</p> <p>7.5 Case studies. Special topics: SOA and Web services, Cloud Computing, Adaptive structures</p>	<b>6</b>	<b>Assignment-7</b>

**Reference Books:**

- Beyond Software architecture, Luke Hohmann, Addison wesley, 2003.
- Software architecture, David M. Dikel, David Kane and James R. Wilson, Prentice Hall PTR, 2001
- Software Design, David Budgen, second edition, Pearson education, 2003
- Head First Design patterns, Eric Freeman & Elisabeth Freeman, O'REILLY, 2007.
- Design Patterns in Java, Steven John Metsker & William C. Wake, Pearson education, 2006
- J2EE Patterns, Deepak Alur, John Crupi & Dan Malks, Pearson education, 2003.
- Design Patterns in C#, Steven John metsker, Pearson education, 2004.
- Pattern Oriented Software Architecture, F. Buschmann & others, John Wiley & Sons
- Anti-Patterns: Refactoring Software, Architectures and Projects in Crisis, William J Brown et al., John Wiley, 1998
- Object-oriented analysis, design and implementation, brahma dathan, sarnathrammath, universities press, 2013
- Design patterns, erich gamma, Richard helan, Ralph johman, john vlissides, PEARSON Publication, 2013.

**Class: M. Sc. (Computer Science) (Sem-III)**

**Paper Code: Comp5305**

**Title of Paper: Lab Course based on Comp5301 & Comp5303**

**Paper: V (Lab Course)**

**Credit: 4 (4 Hours Practical/Week/batch)**

**No. of Practical's: 13**

<b>Sr.No</b>	<b>Practical Assignments</b>
<b>Mobile Technologies Practical Assignments Topics</b>	
<b>1</b>	Assignment based on basic UI Design
<b>2</b>	Assignment based on Basic UI Design
<b>3</b>	Assignment based on Thread and Notification
<b>4</b>	Assignment based on handler on UI thread
<b>5</b>	Assignment based on using Intent, Toast
<b>6</b>	Assignment based on swift in ios
<b>7</b>	Assignment based on startup
<b>8</b>	Assignment based on flutter
<b>Web services Practical Assignments Topics</b>	
<b>1</b>	Introduction of Basic web services
<b>2</b>	Assignment on SOAP
<b>3</b>	Assignment on WSDL
<b>4</b>	Assignment on UDDI
<b>5</b>	Assignment on XML



**Class: M. Sc. (Computer Science) (Sem-III)**

**Title of Paper: Internet of Things (IOT)**

**Credit: 4 (4 Lectures/Week)**

**Pre-Requisite:** Basic understanding of electronics and microprocessors.

**Paper Code: Comp5307**

**Paper: VII (E-III)**

**No. of lectures: 48**

**Course Objectives:**

- The Internet of Things (IoT) is aimed at enabling the interconnection and integration of the physical world and the cyber space.
- To learn about SoC architectures, programming Raspberry Pi and implementation of internet of things and protocols.

**Expected Learning Outcomes:**

- Enable learners to understand System On Chip Architectures.
- Introduction and preparing Raspberry Pi with hardware and installation.
- Learn physical interfaces and electronics of Raspberry Pi and program them using practical's
- Learn how to design IoT based prototypes.

Unit and Contents	No. Of Lectures
<p><b>Unit 1: System on Chip (SoC) and Internet of Things (IoT) Overview</b></p> <ul style="list-style-type: none"><li>- <b>System on Chip:</b> What is System on chip? Structure of System on Chip.</li><li>- <b>SoC products:</b> Field Programmable Gate Array (FPGA), General Purpose Graphics Processing Units (GPU), Accelerated Processing Unit (APU), Compute Units.</li><li>-<b>The IoT paradigm</b> giving overview of IoT supported Hardware platforms such as: Raspberry pi, SoC on ARM 8 Processors, Arduino and Intel Galileo boards.</li><li>-<b>Network Fundamentals:</b> Wired Networking(Router, Switches), Wireless Networking (Access Points)</li><li>-<b>Introduction to Raspberry Pi:</b> Introduction to Raspberry Pi, Raspberry Pi Hardware, Preparing your raspberry Pi.</li><li>-<b>Raspberry Pi Boot:</b> Learn how this small SoC boots without BIOS. Configuring boot sequences and hardware.</li><li>-<b>Introduction to IoT:</b> What is IoT? IoT examples, Simple IoT LED Program.</li><li>-<b>IoT and Protocols</b></li><li>-<b>IoT Security:</b> HTTP, UPnp, CoAP, MQTT, XMPP.</li><li>-<b>IoT Service as a Platform:</b> Clayster, Thinger.io, SenseIoT, carriers and Node RED.</li><li>-<b>IoT Security and Interoperability:</b> Risks, Modes of Attacks, Tools for Security and Interoperability.</li></ul>	20
<p><b>Unit 2: Programming Raspberry Pi</b></p> <p><b>Raspberry Pi and Linux:</b> About Raspbian, Linux Commands, Configuring Raspberry Pi with Linux Commands</p> <p><b>Programing interfaces:</b> Introduction to Node.js, Python.</p> <p><b>Raspberry Pi Interfaces:</b> UART, GPIO, I2C, SPI</p> <p><b>Useful Implementations:</b> Cross Compilation, Pulse Width Modulation, SPI for Camera.</p>	10
<p><b>Unit 3: Case Study &amp; advanced IoT Applications:</b></p> <p>IoT applications in home, infrastructures, buildings, security, Industries, Home appliances, other IoT electronic equipments. Use of Big Data and Visualization in IoT, Industry 4.0 concepts.</p> <p>Sensors and sensor Node and interfacing using any Embedded target boards (Raspberry Pi / Intel Galileo/ARM Cortex/ Arduino)</p>	10
<p><b>Unit 4: Internet of Things Privacy, Security and Governance</b></p> <p>Introduction, Overview of Governance, Privacy and Security Issues, Contribution from FP7 Projects, Security, Privacy and Trust in IoT-Data-Platforms for Smart Cities, First Steps Towards a Secure Platform, Smartie Approach. Data Aggregation for the IoT in Smart Cities, Security</p>	8

**References:**

1. The Internet of Things: From RFID to the Next-Generation Pervasive Networked Lu Yan, Yan Zhang, Laurence T. Yang, Huansheng Ning
2. Internet of Things (A Hands-on-Approach) , Vijay Madiseti , Arshdeep Bahga
3. Designing the Internet of Things , Adrian McEwen (Author), Hakim Cassimally
4. "Mobile Computing," Tata McGraw Hill, Asoke K Talukder and Roopa R Yavagal, 2010.
5. Computer Networks; By: Tanenbaum, Andrew S; Pearson Education Pte. Ltd., Delhi, 4th Edition
6. Data and Computer Communications; By: Stallings, William; Pearson Education Pte. Ltd., Delhi, 6th Edition
7. "Fundamentals of Mobile and Pervasive Computing," F. Adelstein and S.K.S. Gupta, McGraw Hill, 2009.
8. Cloud Computing Bible, Barrie Sosinsky, Wiley-India, 2010