Karmaveer Bhaurao Patil College, Vashi

Autonomous College
Affiliated to University of Mumbai

Syllabus for Approval

Sr. No.	Heading	Particulars
1	Title of Course	Computer Science
2	Eligibility for Admission	3 KT's per SEM
3	Passing marks	40
4	Ordinances/Regulations (if any)	
5	No. of Semesters	Two
6	Level	U.G.
7	Pattern	Choice Based Credit, Grading Semester
8	Status	Revised
9	To be implemented from Academic year	2021-22

AC -

Item No-

Rayat Shikshan Sanstha's

KARMAVEER BHAURAO PATIL COLLEGE, VASHI.
NAVI MUMBAI
(AUTONOMOUS COLLEGE)
Sector-15- A, Vashi, Navi Mumbai - 400 703

Syllabus for S.Y.B.Sc. Computer Science

Program: B.Sc. Computer Science

Course: S.Y.B.Sc. Computer Science

(Choice Based Credit, Grading and Semester System with effect from the academic year 2021-22)

Preamble

The revised and restructured curriculum for the Three-year integrated course is systematically designed considering the current industry needs in terms of skills sets demanded under new technological environment. The proposed curriculum is more contextual, industry affable and suitable to cater the needs of society and nation in present day context.

Second year of this course is about studying core computer science subjects. Theory of Computation course provides an understanding of grammar, syntax and other elements of modern language designs. It also covers developing capabilities to design formulations of computing models and its applications in diverse areas.
The course in Linux Operating System satisfies the need of understanding the structure and functioning of system. Programming holds key indispensable position in any curriculum of Computer Science. It is essential for the learners to know how to use object oriented paradigms(JAVA). There is also one dedicated course Android Developer Fundamentals as a skill enhancement catering to modern day needs of Mobile platforms and applications. The syllabus has Database Systems courses in previous semesters. The course in Database Management Systems is its continuation in third semester.

The course of Combinatorics and Graph Theory in third semester and the course of Linear Algebra in fourth semester take the previous courses in Mathematics. Graph theory is rapidly moving into the mainstream mainly because of its applications in diverse fields which include can further open new opportunities in the areas of genomics, communications networks and coding theory, algorithms and computations and operations research.

Introducing one of the upcoming concepts Physical Computing and IoT programming will definitely open future area as Embedded Engineer, involvement in IoT projects, Robotics and many more. The Raspberry Pi is a popular platform as it offers a complete Linux server in a tiny platform for a very low cost and custom-built hardware with minimum complex hardware builds which is easier for projects in education domain.

Scheme of examination for Each Semester:

Continuous Internal Evaluation: 40 Marks (Common Test-20 Marks & 20 Marks for- Mini Projects, Presentation, Online Course, Case Study, Assignment, Analysis In Statistics, Report Writing, Interviews, Paper Review, Surprise Test).

Semester End Examination: 60 Marks will be as follows-

I.	Theory: The Semester End Examination for theory course work will be conducted as per the following scheme.	
	Each theory paper shall be of two hours duration.	
	All questions are compulsory and will have internal options.	
	Q – I	From Unit – I (having internal options.) 15 M
	Q – II	From Unit – II (having internal options.)15 M
	Q – III	From Unit – III (having internal options.)15 M
	Q – IV	Questions from all the THREE Units with equal weightage of marks allotted to each Unit. 15 M
II.	Practical	The Semester End Examination for practical courses will be conducted as per the following scheme.
Sr. No.	**Particulars of Semester End Practical Examination**	**Marks**
1	Laboratory Work	40
2	Journal	05
3	Viva	05
	TOTAL	**50**

S.Y.B.Sc. Computer Science Syllabus

Credit Based System and Grading System

Academic year 2021-2022

SEMESTER - III

CODE	COURSE TYPE	SUBJECT	SCHEME OF INSTRUCTION (PERIOD PER WEEK)		SCHEME OF EXAMINATION (MAX MARKS)			NO. OF CREDITS
			TH	LAB	CIA	SEE	TOTAL	
UGCS301	CORE	OS & Linux	4	4	40	60	100	3
UGCS302	CORE	Core Java	4	4	40	60	100	3
UGCS303	CORE	PL/SQL	4	4	40	60	100	3
UGCS304	CORE	Combinatorics & Graph Theory	4	3	40	60	100	3
UGCS305	CORE	Computer Network	4	4	40	60	100	3
UGCSP301	CORE SUBJECT PRACTICAL	UGCS301+UGCS302+ UGCS303+UGCS304+ UGCS305					250	5
						TOTAL	750	20

SEMESTER - IV

CODE	COURSE TYPE	SUBJECT	SCHEME OF INSTRUCTION (PERIOD PER WEEK)		SCHEME OF EXAMINATION (MAX MARKS)			NO. OF CREDITS
			TH	LAB	CIA	SEE	TOTAL	
UGCS401	CORE	Advanced JAVA	4	4	40	60	100	3
UGCS402	CORE	Physical Computing and IoT Programming	4	4	40	60	100	3
UGCS403	CORE	Software Engineering	4	3	40	60	100	3
UGCS404	CORE	Skill Enhancement: Android Developer Fundamentals	4	4	40	60	100	3
UGCS405	ELECTIVE	.NET Technologies	4	4	40	60	100	3
UGCSP401	CORE SUBJECT PRACTICAL	UGCS401+UGCS402+ UGCS403+UGCS404+ UGCS405/UGCS406		18			0	5
						TOTAL	1000	20

Note: TH-Theory, CIA- Continuous Internal Assessment, SEE- Semester End Examination.

SEMESTER III

Class: S.Y.B.Sc	**Branch:** Computer Science	**Semester:** III	
Subject: Operating System and Linux			
Period per Week(Each 48 min)	Lecture	04	
	Practical	04	
Evaluation System		**Hours**	**Marks**
	Semester End Examination	2 hrs	60
	Continuous Internal Assessment	__	40
	Semester End Practical Examination	-	-
	Total	__	100

Course: UGCS301	**Operating System and Linux** (Credits : 3 Lectures/Week: 4)	**Lectures**
	Expected Learning Outcomes: After successful completion of this course, students will be able to: 1) Explain the working knowledge of operating System & Linux, from both a graphical and command line perspective, allowing them to easily use any Linux distribution. 2) Apply UNIX/Linux utilities to create and manage simple file processing operations, organize directory structures with appropriate security, and develop shell scripts to perform more complex tasks. 3) Determine as a Developer or Linux System Administrator using the acquired skill set and Identify system performance, network activities. 4) Apply the knowledge of shell scripting and regular expressions**.**	
Unit I	**Introduction and Operating-Systems Structures**: Operating System Structure, Operations and Services; System Calls, Operating-System Design and Implementation; **Process Management:** Process Scheduling and Operations; Interprocess Communication, Process Synchronization, Critical-Section Problem, Peterson's Solution, Semaphores, **CPU Scheduling** – Scheduling criteria, Scheduling algorithms, **Threads -** Overview, Multithreading models, Threading issues	**12 L**
Unit II	**Deadlock -** Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from deadlock.	**12L**

	Memory Management: Contiguous Memory Allocation, Swapping, Paging, Segmentation, Demand Paging, Page Replacement, Allocation of Frames, Thrashing, Memory-Mapped Files. **Storage Management:** Mass-Storage Structure, Disk Structure, Scheduling and Management, RAID Structure.	
Unit III	**Introduction** History of Linux, Philosophy, Community, Terminology, Distributions, Linux kernel vs distribution. **Installation** Hands on Installation using CD/DVD or USB drive. **Linux Structure** Linux Architecture, File System basics, The boot process, init scripts, runlevels, shutdown process, Very basic introductions to Linux processes, Packaging methods: rpm/deb, Graphical Vs Command line. **Graphical Desktop:** Basic Desktop Operations, Text editors: gedit, vi, vim, emacs, Graphics editors, Multimedia applications	**12L**
Unit IV	**Basic bash shell commands:** Starting the shell, Shell prompt, Filesystem Navigation, File and directory listing, File handling, Directory handling, Viewing file contents, Working with data files: Sorting, Searching, Compressing, **Understanding Linux file permission:** Linux security, Using Linux groups, Decoding file permissions, Changing security settings, **The Linux environment variable:** Environment variables, setting environment variables, Removing environment variables, setting the PATH environment variables, Using command aliases.	**12L**
Unit V	**Basic script building:** Using multiple commands, Creating a script file, Displaying messages, Using variables, Redirecting Input and Output, Pipes, Performing math, Exiting the script. **Using structured commands:** Working with the if-then, if-then-else and nesting if statements, test command, the case command, Regular Expressions	**12L**

Textbook(s):
1. Abraham Silberschatz, Peter Galvin, Greg Gagne, Operating System Concepts, Wiley,8th Edition
2. Linux Command line and Shell Scripting: Bible, Richard Blum, Wiley-India.

Additional Reference(s):
1. Achyut S. Godbole, Atul Kahate, Operating Systems, Tata McGraw Hill
2. Unix Concepts and Applications, 4e, Sumitabha Das., TMH.
3. Linux Complete Reference, TMH
4. Official Ubuntu Book, 8th Edit

Links:
1) Linux kernel Home: http://kernel.org
2) Open Source Initiative: https://opensource.org/

3) The Linux Foundation: http://www.linuxfoundation.org/ 4) https://www.geeksforgeeks.org/last-minute-notes-operating-systems/ 5)https://www.tutorialspoint.com/operating_system

Sr. No.	Problems of UGCS301
1	a. Customize desktop environment by changing different default options like changing default background, themes, screensavers b. Screen Resolution: Ascertain the current screen resolution for your desktop c. Networking: Get the current networking configuration for your desktop. Are you on a wired or a wireless connection? What wireless networks are available, if any?
2	Command line operations: a. Install any newpackage on your system b. Remove the package installed c. Find the passwd file in / using find command d. Create an empty file example.txt and move it in /tmp directory using relative pathname. e. Delete the file moved to /tmp in the previous step using absolute path. f. Find the location of ls, ps, bash commands. g. Use man command to find help for various commands
3	Try out the General Purpose Utility Commands
4	Use environment a. Which account are you logged in? How do you find out? b. Display /etc/shadow file using cat and understand the importance of shadow file. How it's different from a passwd file. c. Get your current working directory. d. Explore different ways of getting command history, how to run previously executed commands without typing it? e. Create alias to most commonly used commands like.
5	Linux Editors: Vi Editor a. Create,modify, search, navigate a file in the editor. b. Learn all essential commands like search, search/replace, highlight, show line numbers.
6	Linux Security: a. Uses of root a. Use of sudo command to change user privileges to root b. Identify all operations that require sudo privileges c. Create a new user and add it to sudo configuration file. d. Set password for new user. e. Modify the expiration date for new user using password ageing. f. Delete newly added user.
7	Arithmetic: Write a script which will work as arithmetic calculator to add, subtract, multiply, divide. The user should pass an argument on the command line a letter (a,s,m

	or d) and two numbers. If wrong number of arguments are passed then display an error message. Make use of functions to perform operations
8	Case Statements: Write a script that will be given a month number as the argument and will translate this number into a month name. The result will be printed to stdout.
9	Write a Shell Script that accepts a file name, starting and ending line numbers as arguments and displays all lines between the given line numbers.
10	Programs using awk, sed command

Class: S.Y.B.Sc	**Branch:** Computer Science	**Semester:** III	
Subject: Core Java			
Period per Week(Each 48 min)	Lecture	04	
	Practical	04	
Evaluation System		**Hours**	**Marks**
	Semester End Examination	2 hrs	60
	Continuous Internal Assessment	__	40
	Semester End Practical Examination	-	-
	Total	__	100

Course: UGCS302	**Core Java** (Credits : 2 Lectures/Week: 4)	**Lectures**
	Expected Learning Outcomes: After successful completion of this course, students will be able to: 1) Recall the concept of Object oriented programming using Java & String manipulation. 2) Diagnose the abnormal termination of a java program using exception handling and multithreading. 3) Explain I/O Streams,Networking,Wrapper Classes in Java. 4) Describe the Collection framework, Inner class and AWT.	
Unit I	**Introduction to OOP:** Classes,, Objects, Class Instantiation, Object Initialization, Life cycle of an object, Anonymous object, The four pillars of OOP:Abstraction, Encapsulation, Inheritance, Polymorphism **Introduction to Java:** What is Java?, History, Need and Features of Java, C++ vs Java, Hello World in Java, Java environment setup(installation + path setup)?, JDK, JRE, and JVM (Java Virtual Machine), JVM Memory Management, Internal details of JVM, Z Garbage Collector**,** Unicode System, Operators, Keywords, Data Types, Variables, Naming convention of Java(Camel	**12 L**

	Casing), Constants, Literals and Control Statements like if-else, switch, For loop, while loop, etc. **Practical Approach to OOP Concepts in Java**: Implementing Classes & creating objects Constructor, “this“ keyword, super keyword, implementation of inheritance Simple, Multilevel, Interfaces, Abstract classes and methods, Implementation of Polymorphism, Method Overloading, Method Overriding, Nested and Inner classes, Access Specifiers	
Unit II	**Packages:** Introduction to predefined packages (java.lang, java.util, java.io, java.sql, java.swing), User Defined Packages **Wrapper classes:**Byte, Short, Integer, Long, Float, Double, Character, Boolean classes **Array & String :** Defining an Array, Initializing & Accessing Array, Multi –Dimensional Array, Operation on String, String Buffer, String Tokenizer	**12L**
Unit III	**Exception Handling: Exception:** Exception types, Using try catch and multiple catch, Nested try, throw , throws and finally, Creating user defined Exceptions **Multithreading:** Thread Creation, Thread Life Cycle, Life Cycle Methods Synchronization,- Wait(), -notify(), - notify all() methods **File Handling:** Stream ByteStream Classes CharacterStream Classes, File IO basics, File operations Creating file Reading file(character, byte) Writing file (character, byte)	**12L**
Unit IV	**Networking:** Introduction, Socket, Server socket, Client –Server Communication **Collection:** **Collection Framework, Interfaces:** Introduction, util Package interfaces, List, Set, Map, List interface & its classes, Set interface & its classes, Map interface & its classes	**12L**
Unit V	**Event Handling:** Delegation Event Model, Events, Event classes,Event listener interfaces, Using delegation event model, adapter classes,layouts **Swing:** Need for swing components, Components hierarchy, Panes, Swing components: Jlabel, JTextField **JavaFx**: Introduction to JavaFx, Scene Builder, Basic Controls(TexField, Password, Checkbox, RadioButton, TableView, ListView, FileChooser), Layout Panes, Event Handling.	**12L**

Textbook(s):

1) Herbert Schildt, Java The Complete Reference, Ninth Edition, McGraw-Hill Education, 2014

Additional Reference(s):

1) E. Balagurusamy, Programming with Java, Tata McGraw-Hill Education India, 2014
2) Programming in JAVA, 2nd Ed, Sachin Malhotra & Saurabh Choudhary, Oxford Press

Links:

http://docs.oracle.com/javase/tutorial/
http://www.portcity.edu.bd/ELibrary/CSE/javaprogramming.pdf

Sr. No.	Problems of UGCS301
1	Accept integer values for a, b and c which are coefficients of quadratic equation. Find the solution of quadratic equation
2	Accept two nxn matrices. Write a Java program to find addition of these matrices
3	Accept n strings. Sort names in ascending order.
4	Create a package: Animals. In the package animals create an interface Animal with suitable behaviors. Implement the interface Animal in the same package animals
5	Demonstrate Java inheritance using extends keyword
6	Demonstrate method overloading and method overriding in Java.
7	Demonstrate creating your own exception in Java
8	Design and Develop simple Login Form using Swing and implements Events.
9	Design a Registration form using JavaFx and Implements Event.
10	Design simple calculator GUI application using JavaFX components and implements Event.

Class: S.Y.B.Sc	**Branch:** Computer Science	**Semester:** III	
Subject: PL/SQL			
Period per Week(Each 48 min)	Lecture	04	
	Practical	04	
Evaluation System		**Hours**	**Marks**
	Semester End Examination	2 hrs	60
	Continuous Internal Assessment	__	40
	Semester End Practical Examination	-	-
	Total	__	100

Course: UGCS303	**PL/SQL** (Credits : 2 Lectures/Week: 5)	**Lectures**
	Expected Learning Outcomes: After successful completion of this course, students will be able to: 1. Define the variables, constants, operators and data type of the database system.	

	2. Describe the structure of control statements. 3. Contract the stored procedures and Function in DBMS. 4. Apply Rollback and Commit operations on Database.	
Unit I	**Fundamentals of PL/SQL:** Defining variables and constants **PL/SQL expressions and comparisons:** Logical Operators, Boolean Expressions, CASE Expressions Handling, Null Values in Comparisons and Conditional Statements **PL/SQL Datatypes:** Number Types, Character Types, Boolean Type, Datetime and Interval Types.	**12 L**
Unit II	**Overview of PL/SQL Control Structures:** **Conditional Control:** IF and CASE Statements, IF THEN Statement, IF-THEN-ELSE Statement, IF-THEN-ELSIF Statement, CASE Statement **Iterative Control:** LOOP and EXIT Statements, WHILE-LOOP, FOR-LOOP **Sequential Control:** GOTO and NULL Statements, Concept of nested tables	**12 L**
Unit III	**Stored Procedures:** Types and benefits of stored procedures, creating stored procedures, executing stored procedures, altering stored procedures, viewing stored procedures. PROCEDURE with Parameters (IN,OUT and IN OUT) **Functions:** Difference between Procedures and Functions,User Defined Functions,Nested Functions,Using stored function in SQL statements **Triggers:** Concept of triggers, Implementing triggers - creating triggers, Insert, delete, and update triggers, nested triggers, viewing, deleting and modifying triggers, and enforcing data integrity through triggers,Statement Level Triggers Versus Row Level Triggers, Create Instead of and Disabled Triggers.	**12 L**
Unit IV	**Sequences:** creating sequences, referencing, altering and dropping a sequence. **Cursors :** Implicit Cursor,Explicit Cursor,Cursor Attributes,FOR Loop Cursor statement **Exceptions:** Syntax,RaisingExceptions,User-definedExceptions, Predefined Exceptions	**12 L**
Unit V	**Package:** Package Specification, Package Body, Referring Package Elements,Create Package in PL/SQL ,Forward Declarations,Cursors Usage in Package, Overloading, Dependency in Packages, Package Information. **Collection:** Index-by tables or Associative array, Nested table, Variable-size array or Varray, Constructor and Initialization Concept in Collections, Collection Methods. **Records:** Table-based records, Cursor-based records, User-defined records.	**12 L**

	Transaction: Starting and Ending a Transaction, Committing a Transaction, Rolling Back Transactions,Automatic Transaction Control.	

Textbook(s):

1. Ivan Bayross, “SQL,PL/SQL -The Programming language of Oracle”, B.P.B. Publications
2. Murach’s Oracle SQL and PL SQL by Joel Murach, Murach and Associates.

Additional Reference(s):

1. Oracle PL/SQL Programming, Fifth Edition By Steven Feuerstein, Bill Pribyl
2. Dr. P.S. Deshpande, SQL and PL/SQL for Oracle 10g, Black Book, Dreamtech Press.
3. Oracle Database 11g PL/SQL Programming Workbook, ISBN : 9780070702264, By : Michael McLaughlin, John Harper, Tata McGrawHill
4. Oracle 11g : PL/SQL Reference Oracle Press.

Links:

1) https://docs.oracle.com/database/121/LNPLS/toc.htm
2) https://www.tutorialspoint.com/plsql/index.htm
3) https://www.javatpoint.com/pl-sql-tutorial

Sr. No.	Practicals of UGCS303
1	Write a PL/SQL block for a.Conditional Control b. Iterative Control c. Sequential Control
2	Write a PL/SQL block for a.creating sequences b.Referencing c.altering and dropping a sequence.
3	Write a PL/SQL block for a. Insert/Update/Delete Trigger b. Before/After Trigger c. Working with statement Level Trigger and Row Level Trigger. d. Remove Trigger
4	Write a Procedures in PL/SQL Block (IN, OUT, INOUT, DEFAULT keywords). a. Create an empty procedure, replace a procedure and call procedure b. Create a stored procedure and call it c. Define procedure to insert data d. A forward declaration of procedure
5	Write a Functions in PL/SQL Block. a. Define and call a function b. Define and use function in select clause, c. Call function in dbms_output.put_line d. Recursive function e. Count Employee from a function and return value back f. Call function and store the return value to a variable

6	Write a PL/SQL Block for Cursors a. Cursor attributes:%ROWCOUNT,%FOUND,%NOTFOUND,%ISOPEN b. Cursor with sub queries c. Combination of PL/SQL, cursor and for loop d. Parameterized cursors, Cursor Variables
7	Write a PL/SQL Block for Exception Handling a. Exception Types (implicitly raised, Explicitly raised) b. Trapping Exceptions (WHEN exception1, WHEN OTHERS) c. Predefined Exception – NO_DATA_FOUND – TOO_MANY_ROWS – INVALID_CURSOR – ZERO_DIVIDE – DUP_VAL_ON_INDEX
8	Write a PL/SQL Block for Package.
9	Write a PL/SQL Block for a. Declare and use Association Array b. Varray c. Nested Tables
10	Write a PL/SQL Block to demonstrate the execution of a transaction,Savepoint and Rollback.

Class: S.Y.B.Sc	**Branch:** Computer Science	**Semester:** III	
Subject: Combinatorics & Graph Theory			
Period per Week(Each 48 min)	Lecture	04	
	Practical	03	
Evaluation System		**Hours**	**Marks**
	Semester End Examination	2 hrs	60
	Continuous Internal Assessment	__	40
	Semester End Practical Examination	-	-
	Total	__	100

Course: UGCS304	**Combinatorics & Graph Theory** (Credits : 3 Lectures/Week: 4)	**Lectures**
	Expected Learning Outcomes: After successful completion of this course, students will be able to: 1) Analyze the applications of combinatorics & its uses & Problems , Mathematical Induction. 2) Summarize the concepts of graphs & its different types.	

	3) Select different algorithms to find shortest path & minimal spanning tree. 4) Define the terminology of network flow & discuss the Combinatorial Applications of Network Flows	
Unit I	**Introduction to Combinatorics:** Enumeration, Combinatorics and Graph Theory/ Number Theory/Geometry and Optimization, Sudoku Puzzles. Strings, Sets, and Binomial Coefficients: **Strings-** A First Look, Combinations, Combinatorial, The Ubiquitous Nature of Binomial Coefficients, The Binomial, Multinomial Coefficients **Induction:** Introduction, The Positive Integers are Well Ordered, The Meaning of Statements, Binomial Coefficients Revisited, Solving Combinatorial Problems Recursively, Mathematical Induction, and Inductive Definitions Proofs by Induction. Strong Induction	**12 L**
Unit II	**Graph Theory:** Basic Notation and Terminology of graphs,Walk,Trail,Path,Cycle,Distance between two Pair of Vertices,degree of graph ,Eccentricity of Vertex,Centre & Radius of Graph,Diameter of the graph. Multigraphs: Loops and Multiple Edges, Connected graph, Component,Bridge,Cut set ,Cut Vertex, Edge & Vertex Connectivity, Minimal Degree of graph,Weighted Graph, Isomorphism of graphs, Eulerian and Hamiltonian Graphs, Graph Coloring, Planar Graphs,Clique. Dijikstra's Algorithm **Applying Probability to Combinatorics,** Small Ramsey Numbers, Estimating Ramsey Numbers, Applying Probability to Ramsey Theory, Ramsey's Theorem The Probabilistic Method	**12 L**
Unit III	**Tree :** Basic Notation and Terminology of tree,Eccentricity of Vertex,Centre & Radius of Tree ,Diameter of the Tree, Spanning Tree,Shortest Spanning Tree,Krushkal's Algorithm,Prim's Algorithm, Counting Labeled Trees,Prufer Code.	**12 L**
Unit IV	**Network Flows:** Basic Notation and Terminology, Flows and Cuts, Augmenting Paths, The Ford-Fulkerson Labeling Algorithm, A Concrete Example ,Integer Solutions of Linear Programming Problems.Introduction of Transportation Problem	**12 L**
Unit V	**Combinatorial Applications of Network Flows**: Introduction. Matching in Bipartite Graphs, Poset, Chain partitioning,Permutation group, Pólya's Enumeration Theorem: Coloring the Vertices of a Square	**12 L**
	Textbook(s): 1) Applied Combinatorics, Mitchel T. Keller and William T. Trotter, 2016, http://www.rellek.net/appcomb. **Additional Reference(s):** 1) Applied Combinatorics, sixth.edition, Alan Tucker, Wiley; (2016) 2) Graph Theory and Combinatorics, Ralph P. Grimaldi, Pearson Education; Fifth edition (2012) 3) Combinatorics and Graph Theory, John Harris, Jeffry L. Hirst, Springer(2010).	

	4) Graph Theory: Modeling, Applications and Algorithms, Agnarsson, Pearson Education India (2008). **Links:** 1) http://www.rellek.net/appcomb. 2) http://www.noahc.me/Applied%20Combinatorics%206th%20edition.pdf 3) https://www.geeksforgeeks.org/transportation-problem-set-1-introduction/#:~:text=Transportation%20problem%20is%20a%20special,cost%20of%20transportation%20is%20minimized. 4) http://web.mit.edu/15.053/www/AMP-Chapter-08.pdf	

Sr. No.	**Practicals of UGCS304**
1	Solving problems on strings, sets and binomial coefficients.
2	Solving problems using induction.
3	Solving problems on Eulerian and Hamiltonian graphs
4	Solving problems on Chromatic number and coloring
5	Solving problems using Kruskal's Algorithm
6	Solving problems using Prim's Algorithm
7	Solving problems using Dijkstra's Algorithm
8	Solving problems of finding augmenting paths in network flows.
9	Solving problems on network flows using Ford-Fulkerson Labeling Algorithm
10	Solving problems on posets and their associated networks

Subject: Computer Networks			
Period per Week(Each 48 min)	Lecture	04	
	Practical	04	
Evaluation System		**Hours**	**Marks**
	Semester End Examination	2 hrs	60
	Continuous Internal Assessment	__	40
	Semester End Practical Examination	-	-
	Total	__	100

Course: UGCS305	**Computer Networks** (Credits : 2 Lectures/Week: 4)	**Lectures**
	Expected Learning Outcomes: After successful completion of this course, students will be able to: 1) Explain types of addresses, data communication, OSI model. 2) Examine the concepts of networking, which are important for them to be known as 'networking professionals'. 3) Analyze the concept of networking models, protocols connectionless and connection oriented, functionality of each layer. 4) Interpret routing Algorithms.	
Unit I	**Introduction -** Introduction to data communication, Components, Data Representation, Data Flow, Networks, Network Criteria, Physical Structures, **Network types:** Local Area Network, Wide Area Network, MAN **Network Models**: Protocol layering, OSI Model, TCP/IP Protocol Suite, **Data and Signals**: Analog and Digital Data, Analog and Digital Signals, Wavelength, Time and Frequency Domains, Composite Signals, Bandwidth, Digital Signal, Bit Rate, Bit Length, Transmission of Digital Signals	**12 L**
Unit II	**Transmission Impairments**: Attenuation, Distortion, Noise, Data Rate Limits, Performance, Bandwidth, Throughput, Latency (Delay) **Digital Transmission** – Line Coding (Unipolar, Polar, Biphase), Block Coding(4B/5B Encoding), Analog to digital conversion, PCM, Transmission Modes, Parallel Transmission, Serial Transmission. **Analog Transmission** – Digital to analog conversion (ASK,FSK,PSK, QAM), Analog to Analog conversion - Amplitude Modulation (AM), Frequency Modulation (FM), Phase Modulation (PM),	**12 L**
Unit III	**Multiplexing** – FDM, WDM, Synchronous TDM(time slots & frames, interleaving, data rate management), Spread Spectrum – FHSS, DSSS **Transmission Media** – Guided & Unguided Twisted-Pair Cable, Coaxial Cable, Fiber-Optic Cable, Infrared, Microwaves **Switching** – Switching, Circuit-Switched Networks, Datagram networks, Concept of Virtual circuit networks,	**12 L**
Unit IV	**Data Link Layer** –Error correction & detection, Types of errors, Detection VS Correction, Block Coding, Hamming Distance, Checksum **Multiple Access** – Random(CSMA), Controlled(Reservation, Polling, Token Passing), Channelization(FDMA, TDMA, CDMA) **Connecting Devices** – Repeaters, Hubs, Bridges, Spanning tree algorithm, Two & Three layer Switches, Routers, Gateways, Backbone networks,	**12 L**
Unit V	**IPv4 addresses:** Address Space, Notations, Classful Addressing, NAT **IPv6 addresses.** Transition from IPv4 to IPv6,	**12 L**

	Routing Algorithms: Distance-Vector Routing, Link-State Routing, Path-Vector Routing, **Transport Layer** - UDP, TCP **Congestion Control & Quality of Service**- Data traffic, Congestion, Congestion Control(Open Loop, Closed Loop), QoS and Flow Characteristics **Application Layer** - DNS, SMTP, FTP,	

Textbook(s):
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

Additional Reference(s):
1. Computer Network, Bhushan Trivedi, Oxford University Press
2. Data and Computer Communication, William Stallings, PHI

Links:
1) http://www.crectirupati.com/sites/default/files/lecture_notes/DCN%20NOTES.pdf
2)https://www.tutorialspoint.com/data_communication_computer_network/data_communication_computer_network_tutorial.pdf
3) https://mrcet.com/downloads/digital_notes/ECE/III%20Year/DATA%20COMMUNICATIONS.pdf

Sr. No.	**Practicals of UGCS305**
1	a. Understanding the working of NIC cards, Ethernet/Fast Ethernet/Gigabit Ethernet. b. Study of different types of Network cables and Practically implement the cross-wired cable and straight through cable using crimping tool.
2	Using linux-terminal or Windows-cmd, execute following networking commands and note the output: ping, traceroute, netstat, arp, ipconfig.
3	a. Perform an Initial Switch Configuration b. Perform an Initial Router Configuration
4	Using Packet Tracer, create a basic network of two computers using appropriate network wire
5	Using Packet Tracer, connect multiple (min.6) computers using layer 2 switch
6	Using Packet Tracer create a network using Star Topology
7	Using Packet Tracer create a network using Mesh Topology.
8	Using Packet Tracer, connect a network in triangular shape with three layer two switches and every switch will have four computers. 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

SEMESTER IV

Class: S.Y.B.Sc	**Branch:** Computer Science	**Semester:** IV	
Subject: Advanced Java			
Period per Week(Each 48 min)	Lecture	04	
	Practical	04	
Evaluation System		**Hours**	**Marks**
	Semester End Exam	2 hrs	60
	Continuous Internal Assessment	—	40
	Semester End Practical Examination	—	—
	Total	___	100

Course: UGCS401	**Advanced Java** **(Credits : 2 Lectures/Week: 4)**	**Lectures**
	Expected Learning Outcomes: After successful completion of this course, students will be able to: 1) Describe the JDBC architecture and Perform the connectivity with the database , the servlet , its features and Develop the web application. 2) Discuss the JSP LifeCycle ,its object and Give the various examples on them and JSF as an application with its component. 3) Elaborate the architecture of basic MVC and Struts 2 framework and JavaBeans architecture. 4) Analyze JSON object notation with java.	
Unit I	**JDBC:** Introduction, JDBC Architecture, Types of Drivers, Statement, ResultSet, Read Only ResultSet, Updatable ResultSet, Forward Only ResultSet, Scrollable ResultSet, PreparedStatement, Connection Modes, SavePoint, Batch Updations, CallableStatement, BLOB & CLOB, row sets, metadata, Transaction **Understanding Java EE:** What is an Enterprise Application? What Is Java Enterprise Edition? JavaEETechnologies, JavaEEevolution, Glassfish Server, JavaEE Architecture,Server And Containers: TypesofSystemArchitecture, JavaEEServer, JavaEEContainers.	**12 L**

Unit II	**Introduction to servlets:**Introduction, Web application Architecture, Http Protocol & Http Methods, Web Server & Web Container, why servlets? **Servlet API and Lifecycle: S**ervlet Interface, GenericServlet, HttpServlet, Servlet Life Cycle, ServletConfig, ServletContext, ServletInputStream and ServletOutputStream. Servlet Communication,Cookies, Session Tracking Mechanisms,Request Dispatcher	**12 L**
Unit III	**JSP**: Introduction, JSP LifeCycle, JSP Implicit Objects & Scopes, JSP Directives, JSP Scripting Elements, JSP Actions: Standard actions and customized actions, unified expression language **Java Server Pages Standard Tag Libraries**: What is wrong in using JSPScriptletTags? HowJSTLFixesJSPScriptlet'sShortcomings? Disadvantages Of JSTL, TagLibraries.	**12 L**
Unit-IV	**Java Beans:**Introduction, JavaBeans Properties, **EJB**: Enterprise bean architecture, Benefits of enterprise bean, types of beans, Accessing beans , packaging beans, creating web applications, creating enterprise bean, creating web client, creating JSP file, building and running web applications.	**12 L**
Unit-V	**Struts 2:** Basic MVC Architecture, Struts 2 framework features, Struts 2 MVC pattern, Request life cycle, Examples, Configuration Files, Actions, Interceptors, Results & Result Types, Value Stack/OGNL **JSON:** Overview, Syntax, DataTypes, Objects, Schema, Comparison with XML, JSON with Java	**12 L**
	Textbook(s): 1) 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 4)Java EE 6 for Beginners, Sharanam Shah, Vaishali Shah, Shroff Publishers and Distributors **Additional References(s):** 1) Java EE Project using EJB 3, JPA and struts 2 for beginners, Shah, SPD Java Programming A practical Approach, C Xavier, McGraw Hill Java 2)Server Faces A practical Approach for beginners, B M Harwani, Eastern Economy Edition (PHI). 3)The Java Tutorials of Sun Microsystems Inc 4)Advanced Java Programming, Uttam K. Roy, Oxford University Press	

	Links: **1)** http://docs.oracle.com/javase/tutorial/)	

Sr. No.	Practical of UGCS401
1	Develop simple Registration Form using Swing and stored record and retrieve record in database using JDBC
2	Develop a Java application to store image in a database as well as retrieve image from database.
3	Implement the following Simple Servlet applications. a) Create a simple calculator application using servlet. b) Create a servlet for a login page. If the username and password are correct then it c) says message “Hello <username>” else a message “login failed” d) Create a registration servlet in Java using JDBC. Accept the details such as Username, Password, Email, and Country from the user using HTML Form and store the registration details in the database.
4	Using Request Dispatcher Interface create a Servlet which will validate the password entered by the user, if the user has entered "Servlet" as password, then he will be forwarded to Welcome Servlet else the user will stay on the index.html page and an error message will be displayed.
5	Develop a simple JSP application to pass values from one page to another with validations. (Name-txt, age-txt, hobbies-checkbox, email-txt, gender-radio button).
6	Create Employees table in EMP database. Perform CRUD operations on Employee table using JSP.
7	Create a JSP application to demonstrate the use of JSTL.
8	Develop a Room Reservation System Application Using Enterprise Java Beans.
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.

Class: S.Y.B.Sc	**Branch:** Computer Science	**Semester:** IV	
Subject: Physical Computing and IoT Programming			
Period per Week(Each 48 min)	Lecture	04	
	Practical	04	
Evaluation System		**Hours**	**Marks**
	Semester End Examination	2 hrs	60

	Continuous Internal Assessment	__	40
	Semester End Practical Examination	-	-
	Total	__	100

Course: UGCS402	**Physical Computing and IoT Programming** (Credits : 2 Lectures/Week: 3)	**Lectures**
	Expected Learning Outcomes: After successful completion of this course, students will be able to: 1) Interpret System On Chip Architectures. 2) Prepare Raspberry Pi with hardware and installation. 3) Analyze physical interfaces and electronics of Raspberry Pi and program them using practical's. 4) Examine how to make consumer grade IoT safe and secure with proper use of protocols **.**	
Unit I	**SoC and Raspberry Pi** **System on Chip:** What is System on chip? Structure of System on Chip **SoC products:** FPGA**,** GPU, APU, Compute Units. **ARM 8 Architecture:** SoC on ARM 8. ARM 8 Architecture Introduction **Introduction to Raspberry Pi:** Introduction to Raspberry Pi, Raspberry Pi Hardware, Preparing your raspberry Pi. **Raspberry Pi Boot:** Learn how this small SoC boots without BIOS. Configuring boot sequences and hardware.	**12 L**
Unit II	**Programming Raspberry Pi** **Raspberry Pi and Linux:** About Raspbian, Linux Commands, Configuring Raspberry Pi with Linux Commands **Programing interfaces:** Introduction to Node.js, Python **Raspberry Pi Interfaces:** UART, GPIO, I2C, SPI **Useful Implementations:** Cross Compilation, Pulse Width Modulation, SPI for Camera.	**12 L**
Unit III	**Introduction to IoT:** Sensing, Actuation, Networking basics, Sensor Networks, IoT Definition, Advantages, Disadvantages, IoT Functional Blocks, Physical design of IoT, Logical design of IoT, IOT Technology and Protocols, **IoT applications for industry**: Future Factory Concepts, Smart Objects, Smart Applications, IoT for Retailing Industry, IoT For Oil and Gas Industry, Home Management, eHealth.	**12 L**
Unit IV	**IoT and Protocols** Introduction, Overview of Governance, Privacy and Security Issues **IoT Security:** HTTP, UPnp, CoAP, MQTT, XMPP**.** **IoT Service as a Platform:** Clayster, Thinger.io, SenseIoT, carriots and Node RED. **IoT Security and Interoperability:** Risks, Modes of Attacks, Tools for Security and Interoperability.	**12 L**

Unit V	**Developing IoT solutions:** Introduction to Python, Introduction to different IoT tools, Introduction to Arduino Implementation of IoT with Arduino, Cloud Computing, Fog Computing, Connected Vehicles, Data Aggregation for the IoT in Smart Cities, Simple IoT LED Program.	**12 L**

Textbook(s):

1. Learning Internet of Things, Peter Waher, Packt Publishing(2015)
2. Mastering the Raspberry Pi, Warren Gay, Apress(2014)
3. Vijay Madisetti and Arshdeep Bahga, "Internet of Things (A Hands-on-Approach)", 1 st Edition, VPT, 2014
5. Cuno Pfister, Getting Started with the Internet of Things, O‟Reilly Media, 2011, ISBN: 978-1-4493- 9357-1

Additional Reference(s):

1. Francis daCosta, "Rethinking the Internet of Things: A Scalable Approach to Connecting Everything", 1 st Edition, Apress Publications, 2013
2. Abusing the Internet of Things, Nitesh Dhanjani, O'Reilly

Links:

1) https://www.tutorialspoint.com/internet_of_things/internet_of_things_tutorial.pdf
2) https://www.guru99.com/iot-tutorial.html
3) https://www.raspberrypi.org/magpi-issues/Beginners_Guide_v1.pdf
4)http://www.mosaic-industries.com/embedded-systems/microcontroller-projects/raspberry-pi/microcontroller

Sr. No.	Problems of UGCS402
1	Preparing Raspberry Pi: Hardware preparation and Installation
2	Linux Commands: Exploring the Raspbian
3	GPIO: Light the LED with Python
4	GPIO: LED Grid Module: Program the 8X8 Grid with Different Formulas
5	SPI: Camera Connection and capturing Images using SPI
6	Node RED: Connect LED to Internet of Things
7	Create a simple Web server using Raspberry Pi
8	Stepper Motor Control: PWM to manage stepper motor speed
9	Real Time Clock display using PWM
10	Configure & Test Audio on Raspberry Pi

Class: S.Y.B.Sc	**Branch:** Computer Science	**Semester:** IV	
Subject: Software Engineering			
Period per Week(Each 48 min)	Lecture	04	
	Practical	04	
Evaluation System		**Hours**	**Marks**
	Semester End Exam	2 hrs	60
	Continuous Internal Assessment	—	40
	Semester End Practical Examination	—	—
	Total	___	100

Course: UGCS403	**Software Engineering** **(Credits : 2 Lectures/Week: 5)**	**Lectures**
	Expected Learning Outcomes: After successful completion of this course, students will be able to: 1) Visualize software engineering lifecycle by demonstrating competence in communication, planning, analysis, design, construction, and deployment. 2) Illustrate the techniques and diagrams related to structural modeling 3) Analyze the concept of Project Scheduling and Project Management. 4) correlate the current theories, models, and techniques that provide a basis for the software lifecycle and to use the techniques and tools necessary for engineering practice	
Unit I	**Introduction:** 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 **Software Requirements Analysis and Specification:** Requirements Engineering, Eliciting Requirements, SRS Validation, Components of SRS,Characteristics of SRS.	**12 L**
Unit II	**System Modeling:** 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 **System Design:** 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	**12 L**

Unit III	**Project Scheduling** - Basic Principles, Relationship Between People and Effort, Effort Distribution, Time-Line Charts **Software Measurement and Metrics:** 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	**12 L**
Unit-IV	**Software Project Management: Estimation** in Project Planning Process –Software Scope And Feasibility, Resource Estimation, Empirical Estimation Models – COCOMO II, Estimation for Agile Development, The Make/Buy Decision **Risk Management** - Software Risks, Risk Identification, Risk Projection and Risk Refinement, RMMM Plan	**12 L**
Unit-IV	**Software Configuration Management:** An SCM Scenario, Elements of a Configuration Management System, Baselines Software Configuration Items, Management of Dependencies and Changes **The SCM Repository:** General Features and Content, SCM Features **The SCM Process:** Identification of Objects in the Software Configuration, Version Control, Change Control,Impact Management,Configuration Audit, Status Reporting **Maintenance And Reengineering:** Software Maintenance, Software Supportability,**Reengineering Business Process Reengineering**: Business Processes, A BPR Model, **Software Reengineering:** Software Reengineering Process Model ,Software Reengineering Activities	**12 L**
	Textbook(s): 1) Software Engineering, A Practitioner's Approach, Roger S, Pressman.(2014) **Additional References(s):** 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 **Links:** 1.http://www.resource.mitfiles.com/IT/II%20year/IV%20sem/Software%20Engineering/books/Pressman__Software_Engineering.pdf 2.https://edisciplinas.usp.br/pluginfile.php/2150022/mod_resource/content/1/1429431793.203Software%20Engineering%20by%20Somerville.pdf 3.https://jnec.org/Lab-manuals/CSE/CSE1/BE-Part-1/VM.pdf 4.http://www.startertutorials.com/uml/resources/UML%20and%20DP%20Lab%20Manual.pdf	

Sr. No.	Problems of UGCS403
1	Study the complete Software Development Life Cycle (SDLC) and analyze various activities conducted as a part of various phases. For each SDLC phase, identify the objectives and summaries outcomes.
2	Choose a hypothetical system of significant complexity and write an SRS for the same.
3	Draw one or more Use Case diagrams for capturing and representing requirements of the system.
4	Draw basic class diagrams to identify and describe key concepts like classes, types in your system and their relationships.
5	Draw sequence diagrams OR communication diagrams with advanced notation for your system to show objects and their message exchanges.
6	Draw activity diagrams to display either business flows or like flow charts.
7	Draw component diagrams assuming that you will build your system reusing existing components along with a few new ones.
8	Draw deployment diagrams to model the runtime architecture of your system.
9	Demonstrate the Cyclomatic Complexity.
10	Demonstrate the functional Dependency. Refer: https://www.educba.com/functional-point-analysis/

Class: S.Y.B.Sc	**Branch:** Computer Science	**Semester:** IV	
Subject: Android Developer Fundamentals			
Period per Week(Each 48 min)	Lecture	04	
	Practical	04	
Evaluation System		**Hours**	**Marks**
	Semester End Exam	2 hrs	60
	Continuous Internal Assessment	__	40
	Semester End Practical Examination	__	__
	Total	___	100

Course: UGCS404	**Android Developer Fundamentals** **(Credits : 3 Lectures/Week: 4)**	**Lectures**
	Expected Learning Outcomes: After successful completion of this course, students will be able to:	

	1) Recognize the requirements of the Mobile programming environment. 2) Design and configure Android application development tools ,basic methods and techniques for developing Apps 3) Practice App development on Android Platform and Connect database with App. 4) Develop working prototypes of working systems for various uses in daily lives.	
Unit I	**The Android Platform:** Introduction to the Android platform, Architecture, Android components, Development Tools – SDK, ADB, Gradle, etc. Installing Android Studio IDE, and developing first app **Activities and Lifecycle, Fragments and Intents -** Working with Activities-creating activity, starting activity, managing life cycle of activity, applying themes and styles, displaying dialog in activity; Using Intents-exploring intent objects, resolution, filters passing data using objects in intents; Fragments, Intent Object to Invoke Built-in Application	**12 L**
Unit II	**UI Design:** Display Orientation, Views and ViewGroups, Layouts, Action Bars and Navigation Drawers, Android Layout Managers - LinearLayout, RelativeLayout, ScrollView, TableLayout, FrameLayout, Action Bar, **Working with Views-** TextView, EditText View, Button View, RadioButton View, CheckBox View, ImageButton View, ToggleButton View, RatingBar View, ProgressBar View, AutoCompleteTextView, TimePicker View, DatePicker View, ListView View, Spinner View, ImageView, WebView	**12 L**
Unit III	**Networking in Android:** Accessing the network, Permission to access the network, Checking Network Availability, Sending Email, consuming web services using HTTP **Location-Based Services -** Displaying Maps, Getting Location Data, monitoring a Location, Google Maps API, Using the Geocoder.	**12 L**
Unit-IV	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	**12 L**
Unit-V	**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	**12 L**

	Textbook(s):
	1) “Beginning Android 4 Application Development”, Wei-Meng Lee, March 2012, WROX.
	Additional References(s):
	1)https://developers.google.com/training/courses/android-fundamentals	

	2)https://www.gitbook.com/book/google-developer-training/android-developer-fundamentals-c ourse-practicals/details **Links:** **1) http://www.kmvportal.co.in/Course/MAD/Android%20Book.pdf**	

Sr. No.	**Problems of UGCS404**
1	Install Android Studio and Run Hello World Program.
2	Create an android app that demonstrates Activity Lifecycle and Instance State.
3	Create an android app with Interactive User Interface using Layouts.
4	Create an android app that demonstrates working with TextView Elements.
5	Create an android app that demonstrates the use of Keyboards, Input Controls, Alerts, and Pickers.
6	Develop an application for working with location based services.
7	Create an android app that demonstrates Screen Navigation Using the App Bar and Tabs.
8	Create an android app to Connect to the Internet and use Broadcast Receiver.
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.

Class: S.Y.B.Sc	**Branch:** Computer Science	**Semester:** IV	
Subject: .NET Technologies			
Period per Week(Each 48 min)	Lecture	04	
	Practical	04	
Evaluation System		**Hours**	**Marks**
	Semester End Examination	2 hrs	60
	Continuous Internal Assessment	__	40
	Semester End Practical Examination	-	-
	Total	__	100

Course: UGCS405	**.NET Technologies** (Credits : 3 Lectures/Week: 4)	**Lectures**
	Expected Learning Outcomes: After successful completion of this course, students will be able to: 1) Visualize ADO.NET for data persistence in a web application. 2) Examine the Microsoft .NET Framework and ASP.NET page structure 3) Operate the Master pages, Data binding , LINQ, Rich control,Use page layout, styles, text balance, site map, Master pages and content Pages 4) Create dynamic web pages using c# code, ASP.Net, .MS Visual Studio, NET IDE and Console Applications	
Unit I	**The .NET Framework:**.NET Languages, Common Language Runtime, .NET Class Library **C# Language Basics:** 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 **ASP.NET:** 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,	12
Unit II	**HTML Server Controls** - View State, HTML Control Classes, HTML Control Events, HtmlControl Base Class, HtmlContainerControl Class, HtmlInputControl Class, Page Class, global.asax File, web.config File **Web Controls:** Web Control Classes, WebControl Base Class, List Controls, Table Controls, Web Control Events and AutoPostBack, Page Life Cycle **State Management:** ViewState, Cross-Page Posting, Query String, Cookies, Session State, Configuring Session State, Application State	12
Unit III	**Validation:** Validation Controls, Server-Side Validation, Client-Side Validation, HTML5 Validation, Manual Validation, Validation with Regular Expressions **Rich Controls:** Calendar Control, AdRotator Control, MultiView Control **Themes and Master Pages:** 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 **Website Navigation:** Site Maps, URL Mapping and Routing, SiteMapPath Control, TreeView Control, Menu Control	12
Unit IV	**ADO.NET:** Data Provider Model, Direct Data Access - Creating a Connection, Select Command, DataReader, Disconnected Data Access **Data Binding:** Introduction, Single-Value Data Binding, Repeated-Value Data Binding, Data Source Controls – SqlDataSource **Data Controls:** GridView, DetailsView, FormView	12

	ADO.NET: Data Provider Model, Direct Data Access - Creating a Connection, Select Command, DataReader, Disconnected Data Access	
Unit V	**Data Binding:** Introduction, Single-Value Data Binding, Repeated-Value Data Binding, Data Source Controls – SqlDataSource **Data Controls:** GridView, DetailsView, FormView **Working with XML:** XML Classes – XMLTextWriter, XMLTextReader Caching: When to Use Caching, Output Caching, Data Caching **LINQ:** Understanding LINQ, LINQ Basics, ASP.NET **AJAX:** ScriptManager, Partial Refreshes, Progress Notification, Timed Refreshes	12
	Textbook(s): 1) Beginning ASP.NET 4.5 in C#, Matthew MacDonald, Apress(2012) **Additional References(s):** 1) The Complete Reference ASP .NET, MacDonald, Tata McGraw Hill 2) Beginning ASP.NET 4 in C# and VB Imar Spanajaars, WROX **Links:** 1) https://www.tutorialspoint.com/asp.net/ 2) http://dl.softgozar.com/Files/Ebook/Beginning_ASP.net_4.5.1_in_Csharp_and_VB_Softgozar.com.pdf	

Sr. No.	**Practicals of UGCSP405**
1	Write C# programs for understanding C# basics involving a. Variables and Data Types b. Object-Based Manipulation c. Conditional Logic d. Loops e. Methods
2	Write C# programs for Object oriented concepts of C# such as: a. Program using classes b. Constructor and Function Overloading c. Inheritance d. Namespaces
3	Design ASP.NET Pages with a. Server controls. b. Web controls and demonstrate the use of AutoPostBack c. Rich Controls (Calendar / Ad Rotator)
4	Design ASP.NET Pages for State Management using a. Cookies b. Session State c. Application State
5	Perform the following activities a. Design ASP.NET page and perform validation using various Validation Controls 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.