Scheme & Syllabus of

Bachelor of Technology

Computer Science & Engineering

Batch 2019 onwards

(5th Semester)

By

Department of Academics

IK Gujral Punjab Technical University

Bachelor of Technology in Computer Science & Engineering

It is a Graduate (UG) Programme of 4 years duration (8 semesters)

Courses & Examination

Fifth Semester

Course Code	Type of Course	Course Title	Hours per Week			Marks Distribution		Total Marks	Credits
			L	T	P	Internal	External		
BTCS 501-18	Professional Core Courses	Database Management Systems	3	0	0	40	60	100	3
BTCS 502-18	Professional Core Courses	Formal Language & Automata Theory	3	1	0	40	60	100	3
BTCS 503-18	Professional Core Courses	Software Engineering	3	0	0	40	60	100	3
BTCS 504-18	Professional Core Courses	Computer Networks	3	0	0	40	60	100	3
BTCS XXX-18	Professional Elective	Elective-I	3	0	0	40	60	100	3
BTCS YYY-18	Professional Elective Courses	Elective-II	3	0	0	40	60	100	3
BTCS 505-18	Professional Core Courses	Database Management Systems Lab	0	0	4	30	20	50	2
BTCS 506-18	Professional Core Courses	Software Engineering Lab	0	0	2	30	20	50	1
BTCS 507-18	Professional Core Courses	Computer Networks Lab	0	0	2	30	20	50	1
BTCS XXX-18	Professional Elective	Elective-I Lab	0	0	2	30	20	50	1
BTCS YYY-18	Professional Elective Courses	Elective-II lab	0	0	2	30	20	50	1
		Industrial Training	0	0	0	60	40	100	Satisfactory/Un satisfactory
Total			**18**	**1**	**12**	**450**	**500**	**950**	**24**

LIST OF ELECTIVES

Elective-I

BTCS 508-18 Programming in Java

BTCS 509-18 Web and Open Source Technologies

BTCS 510-18 Programming in Python

BTCS 511-18 Programming in Java lab

BTCS 512-18 Web and Open Source Technologies lab

BTCS 513-18 Programming in Python Lab

Elective-II

BTCS 514-18 Mobile Application Development

BTCS 515-18 Computer Graphics

BTCS 516-18 Internet of Things

BTCS 517-18 Mobile Application Development lab

BTCS 518-18 Computer Graphics Lab

BTCS 519-18 Internet of Things Lab

Fifth Semester

Course Code: BTCS501-18	Course Title: Database Management Systems	3L:0T:0P	3 Credits

Detailed contents

Module 1: Database system architecture: Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML). Data models: Entity-relationship model, network model, relational and object oriented data models, integrity constraints, data manipulation operations. **[7hrs](CO 1, 2)**

Module 2: Relational query languages: Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS - MYSQL, ORACLE, DB2, SQL server. Relational database design: Domain and data dependency, Armstrong's axioms, Normal forms, Dependency preservation, Lossless design. Query processing and optimization: Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms.

[10hrs](CO 2, 4)

Module 3: Storage strategies: Indices, B-trees, hashing. **[3hrs](CO 3)**

Module 4: Transaction processing: Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp based schedulers, Multi-version and optimistic Concurrency Control schemes, Database recovery. **[6hrs](CO 5, 6)**

Module 5: Database Security: Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection. **[8hrs](CO 4, 5)**

Module 6: Advanced topics: Object oriented and object relational databases, Logical databases, Web databases, Distributed databases . **[8hrs](CO 4, 6)**

Course Outcomes:

At the end of the course the student should be able to:

CO 1: For a given query write relational algebra expressions for that query and optimize the developed expressions

CO 2: For a given specification of the requirement design the databases using ER method and normalization.

CO 3: For a given specification construct the SQL queries for Open source and Commercial DBMS - MYSQL, ORACLE, and DB2.

CO 4: For a given query optimize its execution using Query optimization algorithms

CO 5: For a given transaction-processing system, determine the transaction atomicity, consistency, isolation, and durability.

CO 6: Implement the isolation property, including locking, time stamping based on concurrency control and Serializability of scheduling.

Suggested books:

1. **"Database System Concepts"**, 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.

Suggested reference books

1 **"Principles of Database and Knowledge – Base Systems"**, Vol 1 by J. D. Ullman, Computer Science Press.

2 **"Fundamentals of Database Systems"**, 5th Edition by R. Elmasri and S. Navathe, Pearson Education.

3 **"Foundations of Databases"**, Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley.

Course Code : BTCS 502-18	**Course Title: Formal Language & Automata Theory**	**3L:1T:0P**	**3Credits**

Detailed contents

Module 1: Introduction: Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages. Regular languages and finite automata: Regular expressions and languages, deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages, minimization of finite automata. Context-free languages and pushdown automata: Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, nondeterministic pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs. Context-sensitive languages: Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG. Turing machines: The basic model for Turing machines (TM), Turing recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators. Undecidability: Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice s theorem, undecidable problems about languages.

[4hrs](CO 1)

Module 2: Regular languages and finite automata: Regular expressions and languages, deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages, minimization off finite automata.

[8hrs](CO 2, 3)

Module 3: Context-free languages and pushdown automata Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, nondeterministic pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs. **[8hrs](CO 4, 5)**

Module 4: Context-sensitive languages Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG. **[6hrs](CO 5)**

Module 5: Turing machines The basic model for Turing machines (TM), Turing recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators. **[8hrs](CO 5.6)**

Module6: Undecidability Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rices theorem, undecidable problems about languages. **[8hrs](CO 7)**

Course Outcomes:

At the end of the course the student should be able to:

CO 1: Write a formal notation for strings, languages and machines.

CO 2: Design finite automata to accept a set of strings of a language.

CO 3: For a given language determine whether the given language is regular or not.

CO 4: Design context free grammars to generate strings of context free language.

CO 5: Determine equivalence of languages accepted by Push Down Automata and languages generated by context free grammars

CO 6: Write the hierarchy of formal languages, grammars and machines.

CO 7: Distinguish between computability and non-computability and Decidability and undecidability.

Suggested books

1. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, **Introduction to Automata Theory, Languages, and Computation**, Pearson Education Asia.

Suggested reference books:

1. Harry R. Lewis and Christos H. Papadimitriou, **Elements of the Theory of Computation**, Pearson Education Asia.
2. Dexter C. Kozen, **Automata and Computability**, Undergraduate Texts in Computer Science, Springer.
3. Michael Sipser, **Introduction to the Theory of Computation**, PWS Publishing.
4. John Martin, **Introduction to Languages and the Theory of Computation**, Tata McGraw Hill.

Course Code: BTCS 503-18	Course Title : Software Engineering	3L:0T:0P	3 Credits

Detailed Contents:

UNIT 1: Evolution and impact of Software engineering, software life cycle models: Waterfall, prototyping, Evolutionary, and Spiral models. Feasibility study, Functional and Non-functional requirements, Requirements gathering, Requirements analysis and specification. [**8hrs] (CO 1)**

UNIT 2: Basic issues in software design, modularity, cohesion, coupling and layering, function-oriented software design: DFD and Structure chart, object modeling using UML, Object-oriented software development, user interface design. Coding standards and Code review techniques.

[6hrs] (CO 2)

UNIT 3: Fundamentals of testing, White-box, and black-box testing, Test coverage analysis and test case design techniques, mutation testing, Static and dynamic analysis, Software reliability metrics, reliability growth modeling. [**8 hrs] (CO 3)**

UNIT 4: Software project management, Project planning and control, cost estimation, project scheduling using PERT and GANTT charts, cost-time relations: Rayleigh-Norden results, quality management [**8 hrs] (CO 4)**

UNIT 5: ISO and SEI CMMI, PSP and Six Sigma. Computer aided software engineering, software maintenance, software reuse, Component-based software development. [**6 hrs] (CO 5)**

Course Outcomes:

At the end of the course the student should be able to:

CO 1: Students should be able to identify the need for engineering approach to software development and various processes of requirements analysis for software engineering problems.

CO 2: Analyze various software engineering models and apply methods for design and development of software projects.

CO 3: Work with various techniques, metrics and strategies for testing software projects.

CO 4: Identify and apply the principles, processes and main knowledge areas for Software Project Management

CO 5: Proficiently apply standards, CASE tools and techniques for engineering software projects

Suggested Readings/ Books:

1. Roger Pressman, **"Software Engineering: A Practitioners Approach,** (6th Edition), McGraw Hill,
1. 1997.
2. Sommerville, **"Software Engineering, 7th edition**", Adison Wesley, 1996.
3. Watts Humphrey, **"Managing software process**", Pearson education, 2003.
4. James F. Peters and Witold Pedrycz, **"Software Engineering – An Engineering Approach"**, Wiley.
5. Mouratidis and Giorgini. **"Integrating Security and Software Engineering–Advances and Future"**, IGP. ISBN – 1-59904-148-0.
6. Pankaj Jalote, **"An integrated approach to Software Engineering"**, Springer/Narosa.

Course Code: BTCS 504 -18	Course Title: Computer Networks	3L:0T:0P	3Credits

<u>Detailed Contents:</u>

Module 1: Data Communication Components

Representation of data and its flow Networks, Various Connection Topology, Protocols and Standards, OSI model, Transmission Media, LAN: Wired LAN, Wireless LANs, Connecting LAN and Virtual LAN, Techniques for Bandwidth utilization: Multiplexing- Frequency division, Time division and Wave division, Concepts on spread spectrum. **[8 hrs] (CO 1)**

Module 2: Data Link Layer and Medium Access SubLayer

Error Detection and Error Correction- Fundamentals, Block coding, Hamming Distance, CRC; Flow Control and Error control protocols-Stop and Wait, Goback–NARQ, Selective Repeat ARQ, Sliding Window, Piggy backing, Random Access, Multiple access protocols- Pure ALOHA, Slotted ALOHA, CSMA/CDCDMA/CA. **[10 hrs] (CO 2)**

Module 3: Network Layer

Switching, Logical addressing – IPV4, IPV6; Address mapping – ARP, RARP, BOOTP and DHCP – Delivery, Forwarding and Unicast Routing protocols. **[8 hrs] (CO 3)**

Module 4: Transport Layer

Process to Process Communication, User Datagram Protocol(UDP), Transmission Control Protocol (TCP), SCTP Congestion Control; Quality of Service, QoS improving techniques: Leaky Bucket and Token Bucket algorithm. **[8 hrs] (CO 3)**

Module 5: Application Layer

Domain Name Space (DNS), DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls, Basic concepts of Cryptography. **[8 hrs] (CO 4)**

<u>Course Outcomes:</u>

The student will be able to:
CO 1: Explain the functions of the different layer of the OSI Protocol
CO 2: Describe the function of each block of wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs);
CO 3: Develop the network programming for a given problem related TCP/IP protocol
CO 4: Configure DNS DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls using open source available software and tools.

<u>SuggestedBooks</u>

1. **Data Communication and Networking**, 4th Edition, Behrouz A. Forouzan, McGraw-Hill.
2. **Data and Computer Communication**, 8th Edition, William Stallings, Pearson Prentice Hall India.

<u>Reference Books</u>

1. **Computer Networks**, 8th Edition, Andrew S. Tanenbaum, Pearson New International Edition.
2. **Internet working with TCP/IP**, Volume1, 6th Edition Douglas Comer, Prentice Hall of India.
3. **TCP/IP Illustrated**, Volume1, W. Richard Stevens, Addison-Wesley, United States of America.

Elective-I

Course Code: BTCS 508-18	Course Title: Programming in JAVA	L:3 ;T:0; P:0	3 Credits

Detailed Contents:

Unit 1:

Overview: Object oriented programming principles, Java essentials, java virtual machine, program structure in java

Java class libraries, Data types, Variables and Arrays, Data types and casting, automatic type promotion in expressions, arrays.

Operators and Control Statements: Arithmetic operators, bit wise operators, relational operators, Boolean logical operators, the? Operator, operator precedence

Java's selection statements, iteration statements, jump statements. **CO 1**

UNIT 2:

Introduction to Classes: Class fundamentals, declaring class, creating objects

Introducing methods: method declaration, overloading, using objects as parameters, recursion

Constructors, this keyword, garbage collection, the finalization **CO 1**

UNIT 3:

Inheritance: Inheritance basics, using super and final, method overriding, dynamic method dispatch, Abstract Class

Interface: variables and extending Interfaces

Package: Creating and importing packages, Package access protection,

Exception Handling: Exception handling fundamentals, Exception types, Uncaught Exceptions Using try and catch, multiple catch clauses, nested try statements, throw, Java's built-in exceptions. **CO 1,2**

UNIT 4:

Multithreaded Programming: The Java thread model, the main thread, creating thread, creating multiple threads, using is Alive () and join (), Thread priorities, synchronization, Inter thread communications, suspending resuming and stopping threads. **CO 3**

UNIT5:

I/O: I/O Basics, Reading Console Input, Writing Console Output, Reading and Writing Files

Applets: Applet Fundamentals, Applet Architecture, The HTML Applet tag, Passing parameters to Applets.

Networking: Networking basics, Java and the Net, TCP/IP Client Sockets URL, URL Connection, TCP/IP Server Sockets, Database connectivity. **CO 4**

Course Outcomes:

At the end of the course the student should be able to:

CO1: Understand the features of Java such as opeartors, classes, objects, inheritance, packages and exception handling

CO2: Learn latest features of Java like garbage collection, Console class, Network interface, APIs

CO3: Acquire competence in Java through the use of multithreading, applets

CO4: Get exposure to advance concepts like socket and database connectivity.

Suggested Readings/Books

1. Herbert Schildt, **The Complete Reference Java 2**, McGraw-Hill.
2. Joyce Farrell, **Java for Beginners**, Cengage Learning.
3. Deitel and Deitel, **Java: How to Program**, 6th Edition, Pearson Education.
4. James Edward Keogh, Jim Keogh, J2EE: **The complete Reference**, Mc Graw Hill
5. Khalid A. Mughal, Torill Hamre, Rolf W. Rasmussen, **Java Actually**, Cengage Learning.
6. Shirish Chavan, **Java for Beginners**, 2nd Edition, Shroff Publishers.

Course Code: BTCS 509-18	Course Title: Web and Open Source Technologies	L:3; T:0; P:0	3 Credits

Detailed Syllabus:

Introduction to WWW: Protocols and programs, secure connections, application and development tools, the web browser, Server, choices, setting up UNIX and Linux web servers, Logging users, dynamic IP

Web Design: Web site design principles, planning the site and navigation

Introduction to HTML: The development process, Html tags and simple HTML forms, web site structure

Introduction to XHTML: XML, Move to XHTML, Meta tags, Character entities, frames and frame sets, inside browser.

Style sheets : Need for CSS, introduction to CSS, basic syntax and structure, using CSS, background images, colors and properties, manipulating texts, using fonts, borders and boxes, margins, padding lists, positioning using CSS, CSS2

JavaScript: Client side scripting, Javascript, How to develop Javascript, simple Javascript, variables, functions, conditions, loops and repetition.

Advance script, Javascript and objects, Javascript own objects, the DOM and web browser environments, forms and validations

DHTML: Combining HTML, CSS and Javascript, events and buttons, controlling your browser

CO 1

Ajax: Introduction, HTTP request, XHttpResponse, AJAX Server Script, AJAX Database, Advantages & disadvantages, Purpose of it, Ajax based web application, alternatives of Ajax

XML: Introduction to XML, uses of XML, simple XML and XML key components, DTD and Schemas, Well formed, using XML with application.XML, XSL and XSLT. Introduction to XSL, XML transformed simple example, XSL elements, transforming with XSLT **CO 2**

PHP: Starting to script on server side, syntax, statements, operators, Arrays, function and forms sessions, E-mail, PHP and AJAX, advance PHP

MySQL Databases : Basic command with PHP examples, Connection to server, creating database, selecting a database, listing database, listing table names creating a table, inserting data, altering tables, queries, deleting database, deleting data and tables, PHPmyadmin and database bugs.

JavaScript Library & Web-Framework:

Jquery: Introduction, Why jQuery, jQuery methods for DOM manipulation, jQuery methods for CSS manipulation, jQuery AJAX Methods (Asynchronous JavaScript and XML)

AngularJS: Fundamental structural concepts of AngularJS, AngularJS Directives, AngularJS Expressions, Use of custom attributes in HTML, introduction to modules and controllers, form validation using validation rules, Server Communication & Data Binding techniques. **CO 3**

Course Outcomes:

At the end of the course the student should be able to:
CO 1: Students are able to develop a dynamic webpage by the use of java script and DHTML.
CO 2: Students will be able to write a well formed / valid XML document.

CO 3: Students will be able to write a server side java application called JSP to catch form data sent from client and store it on database

Suggested Readings/Books:

1. Deitel, Deitel, Nieto, and Sandhu: **XML How to Program**, Pearson Education.
2. Herbert Schildt: **Java 2: The Complete Reference**, Fifth Edition, TMH.
3. Ivan Bayross: **Web Enabled Commercial Application.**
4. Schafer: **Development**, BPB.
5. **HTML, CSS, Java Script,Perl, Python and PHP**, Wiley India Textbooks.
6. R. Peterson, 2007, **Linux: The Complete Reference**, Sixth Edition, TMH.

Course Code: BTCS 510-18	Course Title: Programming in Python	L: 3; T: 0; P:0	3 Credits

Detailed Syllabus:

UNIT - I Python Basics, Objects- Python Objects, Standard Types, Other Built-in Types, Internal Types, Standard Type Operators, Standard Type Built-in Functions, Categorizing the Standard Types, Unsupported Types Numbers - Introduction to Numbers, Integers, Floating Point Real Numbers, Complex Numbers, Operators, Built-in Functions, Related Modules Sequences - Strings, Lists, and Tuples, Mapping and Set Types **CO 1,2**

UNIT - II FILES: File Objects, File Built-in Function [open()], File Built-in Methods, File Built-in Attributes, Standard Files, Command-line Arguments, File System, File Execution, Persistent Storage Modules, Related Modules Exceptions: Exceptions in Python, Detecting and Handling Exceptions, Context Management, *Exceptions as Strings, Raising Exceptions, Assertions, Standard Exceptions, *Creating Exceptions, Why Exceptions (Now)?, Why Exceptions at All?, Exceptions and the sys Module, Related Modules Modules: Modules and Files, Namespaces, Importing Modules, Importing Module Attributes, Module Built-in Functions, Packages, Other Features of Modules **CO 2,3**

UNIT - III Regular Expressions: Introduction, Special Symbols and Characters, Res and Python Multithreaded Programming: Introduction, Threads and Processes, Python, Threads, and the Global Interpreter Lock, Thread Module, Threading Module, Related Modules **CO 3,4**

UNIT - IV GUI Programming: Introduction, Tkinter and Python Programming, Brief Tour of Other GUIs, Related Modules and Other GUIs WEB Programming: Introduction, Wed Surfing with Python, Creating Simple Web Clients, Advanced Web Clients, CGI-Helping Servers Process Client Data, Building CGI Application Advanced CGI, Web (HTTP) Servers **CO 4,5**

UNIT – V Database Programming: Introduction, Python Database Application Programmer's Interface (DB-API), Object Relational Managers (ORMs), Related Modules **CO 5**

Course Outcomes:

At the end of the course the student should be able to:
CO 1: Examine Python syntax and semantics and be fluent in the use of Python flow control and functions.

CO 2: Demonstrate proficiency in handling Strings and File Systems.

CO 3: Create, run and manipulate Python Programs using core data structures like Lists, Dictionaries and use Regular Expressions.

CO 4: Interpret the concepts of Object-Oriented Programming as used in Python.

CO 5: Implement exemplary applications related to Network Programming, Web Services and Databases in Python.

Suggested Readings/Books

1. Textbook 1**. Core Python Programming**, Wesley J. Chun, Second Edition, Pearson.

Elective-II

Course Code: BTCS 514-18	Course Title: Mobile Application Development	L:3;T:0; P:0	3 Credits

Detailed Syllabus:

Unit-1

Introduction to Android: The Android Developing environment, Android SDK, Introduction to Open Handset Alliance, Development Framework, Application Fundamentals; Device Compatibility, System permissions, Understanding Anatomy of Android Application, Android Development Tools **CO 1**

Unit-II

Getting started with Mobility: Mobility Landscape, Mobile Platforms, Mobile apps development, Android terminologies, Application Context, Activities, Services, Intents, Receiving and Broadcasting Intents, Setting up the mobile apps development environment with emulator **CO 1, 2**

Unit-III

Building block of Mobile apps: App user Interface Designing, Layout, User Interface elements, VUIs and Mobile Apps, Text to Speech Techniques, Designing the Right UI, Activity states and lifecycle, Interaction among activities **CO 2,3**

Unit-IV

Sprucing up Mobile apps: App functionality beyond user interface- Threads, sync task, Services-states and life cycle, Notifications, Broadcast receivers, Telephony and SMS APIs Native data handling: on device file I/O, shared preferences, mobile databases such as SQLite, Working with a content provider **CO 3,4**

Unit-V

Factors in Developing Mobile Applications: Mobile Software Engineering, Frameworks and Tools, Generic UI Development, Android User

Graphics and Multimedia: Performance and Multithreading, Graphics and UI Performance, Android Graphics, Mobile Agents and Peer-to-Peer Architecture, Android Multimedia **CO 4,5**

Unit-VI

Platforms and Additional Issues: Development Process, Architecture, Design, Technology Selection, Testing, Security and Hacking, Active Transactions, More on Security **CO 4**

Unit-VII

Deployment of apps: Versioning, signing and packaging mobile apps, distributing apps on market place. **CO 5**

Course Outcomes:

At the end of the course the student should be able to:
CO 1: Describe those aspects of mobile programming that make it unique from programming for other platforms,

CO 2: Critique mobile applications on their design pros and cons,

CO 3: Utilize rapid prototyping techniques to design and develop sophisticated mobile interfaces,
CO 4: Program mobile applications for the Android operating system that use basic and advanced phone features, and
CO 5: Deploy applications to the Android marketplace for distribution

References:

1. Rick Rogers, John Lombardo, Meike Blake, "**Android application development**", Ist Edition, O'Reilly, 2010.
2. T1.Lauren Darcey and Shane Conder,"**Android Wireless Application Development**", 2nd ed. Pearson Education, 2011.
3. Wei-Meng Lee , **Beginning Android 4 development**, 2012 by John Wiley & Sons
4. Jeff Mewherter, Scott Gowell,WroxPublisher,"**Professional Mobile Application Development**", Ist Edition, 2012.
5. Reto Meier, "**Professional Android 4 Application Development**", Wrox, 2012.

Course Code: BTCS 515-18	Course Title: Computer Graphics	L:3; T:0; P:0	3 Credits

Detailed Syllabus:

UNIT–I

Overview of Computer Graphics: Basics of Computer Graphics, Applications, Video Display devices, Raster–Scan displays, Random–Scan displays, Color CRT Monitors, Flat–Panel Displays; Video Controller, Display Processor, Common Graphic Input and Output devices, Graphic File Formats, Graphics Software's. **CO 1**

Unit- II

Output Primitives: Line Drawing, DDA, Bresenham Line Algorithm; Mid-Point Line Algorithm, Bresenham Circle Algorithm, Midpoint Circle drawing algorithms; Midpoint Ellipse Algorithm; Flood and Boundary Filling. **CO 2**

Unit- III

Two-Dimensional Geometric Transformation: Translation, Rotation, Scaling, Reflection, Shearing, Matrix representations; Composite transformations. **CO 2**

UNIT–IV

Two-Dimensional Viewing: Viewing coordinate reference frame; Window to Viewport coordinate transformation. Point Clipping, Line Clipping, text Clipping; Cohen–Sutherland and Liang–Barskey Algorithms for line clipping; Sutherland–Hodgeman algorithm for polygon clipping.

CO 3, 4

Unit- V

Three Dimensional Transformations & Viewing: Translation, Rotation, Scaling, Reflection and composite transformations. Parallel and Perspective Projections, Viewing Transformation: View Plan, View Volumes and Clipping. **CO 4, 5**

Unit- VI

3 D Graphics and Visibility: Plane projections and its types, Vanishing points, Specification of a 3D view. Image and object precision, Hidden edge/surface removal or visible edge/surface determination techniques; z buffer algorithms, Depth sort algorithm, Scan line algorithm and Floating horizon technique. **CO 5**

Unit –VII

Color Models: Properties of Light, Intuitive Color Concepts, concepts of chromaticity, RGB Color Model, CMY Color Model, HLS and HSV Color Models, Conversion between RGB and CMY color Models, Conversion between HSV and RGB color models, Color Selection and Applications. **CO 5, 6**

UNIT–VIII

Animation: Graphics Design of Animation sequences, General Computer Animation Functions Introduction to Rendering, Raytracing, Antialiasing, Fractals, Gourard and Phong shading. **CO 6**

Course Outcomes:

At the end of the course the student should be able to:

CO 1. To list the concepts used in computer graphics.

CO 2. To implement various algorithms to scan, convert the basic geometrical primitives, transformations, Area filling, clipping.

CO 3. To describe the importance of viewing and projections.

CO 4. To define the fundamentals of animation, virtual reality and its related technologies.

CO 5. To understand a typical graphics pipeline

CO 6. To design an application with the principles of virtual reality

References:

1. D. Hearn and M.P. Baker, **Computer Graphics: C version**, 2nd Edition, PHI, 2004.
2. D.F. Rogers, **Procedural Elements for Computer Graphics**, 2nd Edition, Addison Wasley, 2004.
3. D.F. Rogers, **Mathematical Elements for Graphics**, 2nd Edition. McGraw Hill, 2004.
4. J.D. Foley et al, **Computer Graphics, Principles and Practices**, 2nd Edition, Addison Wasley, 2004.
5. Roy A. Plastock, Gordon Kalley, **Computer Graphics**, Schaum’s Outline Series, 1986.

Course Code: BTCS 516-18	**Course Title:Internet of Things**	**L:3; T:0; P:0**

Detailed Syllabus:

1. Introduction to IoT (8 Hours)

Architectural Overview, Design principles and needed capabilities, IoT Applications, Sensing, Actuation, Basics of Networking, M2M and IoT Technology Fundamentals- Devices and gateways, Data management, Business processes in IoT, Everything as a Service(XaaS), Role of Cloud in IoT, Security aspects in IoT. **CO 1**

2. Elements of IoT (9 Hours)

Hardware Components- Computing (Arduino, Raspberry Pi), Communication, Sensing, Actuation, I/O interfaces.

Software Components- Programming API's (using Python/Node.js/Arduino) for Communication CO2

Protocols-MQTT, ZigBee, Bluetooth, CoAP, UDP, TCP. **CO 2**

3. IoT Application Development (18 Hours)

Solution framework for IoT applications- Implementation of Device integration, Data acquisition and integration, Device data storage- Unstructured data storage on cloud/local server, Authentication, authorization of devices. **CO 3**

4. IoT Case Studies (10 Hours)

IoT case studies and mini projects based on Industrial automation, Transportation, Agriculture, Healthcare, Home Automation **CO 4**

Course Outcomes:

At the end of the course the student should be able to:

CO 1. To understand internet of Things and its hardware and software components

CO 2. To develop an Interface I/O devices, sensors & communication modules

CO 3. To remotely monitor data and control devices

CO 4. To develop real life IoT based projects

LIST OF SUGGESTED BOOKS

1. Vijay Madisetti, Arshdeep Bahga, **Ïnternet of Things**, "A Hands on Approach", University Press.
2. Dr. SRN Reddy, Rachit Thukral and Manasi Mishra, "**Introduction to Internet of Things: A practical Approach**", ETI Labs.
3. Pethuru Raj and Anupama C. Raman, "**The Internet of Things: Enabling Technologies, Platforms, and Use Cases**", CRC Press.
4. Jeeva Jose, "**Internet of Things**", Khanna Publishing House, Delhi.
5. Adrian McEwen, "**Designing the Internet of Things**", Wiley.
6. Raj Kamal, "**Internet of Things: Architecture and Design**", McGraw Hill.
7. Cuno Pfister, "**Getting Started with the Internet of Things**", O Reilly Media.

Course Code: BTCS 505-18	**Course Title: Database management System lab**	**0L:0T:4P**	**2 Credits**

Detailed List of Tasks:

1. Introduction to SQL and installation of SQL Server / Oracle.
2. Data Types, Creating Tables, Retrieval of Rows using Select Statement, Conditional Retrieval of Rows, Alter and Drop Statements.
3. Working with Null Values, Matching a Pattern from a Table, Ordering the Result of a Query, Aggregate Functions, Grouping the Result of a Query, Update and Delete Statemets.
4. Set Operators, Nested Queries, Joins, Sequences.
5. Views, Indexes, Database Security and Privileges: Grant and Revoke Commands, Commit and Rollback Commands.
6. PL/SQL Architecture, Assignments and Expressions, Writing PL/SQL Code, Referencing Non-SQL parameters.
7. Stored Procedures and Exception Handling.
8. Triggers and Cursor Management in PL/SQL.

Suggested Tools – MySQL, DB2, Oracle, SQL Server 2012, Postgre SQL, SQL lite

Course Outcomes:

CO1: This practical will enable students to retrieve data from relational databases using SQL.

CO2: students will be able to implement generation of tables using datatypes

CO3: Students will be able to design and execute the various data manipulation queries.

CO4: Students will also learn to execute triggers, cursors, stored procedures etc.

Course Code: BTCS506-18	Course Title: Software Engineering Lab	L:0;T:0; P:2	1Credits

Detailed List of Tasks:

1. Study and usage of OpenProj or similar software to draft a project plan
2. Study and usage of OpenProj or similar software to track the progress of a project
3. Preparation of Software Requirement Specification Document, Design Documents and Testing Phase
4. related documents for some problems
5. Preparation of Software Configuration Management and Risk Management related documents
6. Study and usage of any Design phase CASE tool
7. To perform unit testing and integration testing
8. To perform various white box and black box testing techniques
9. Testing of a web site

Suggested Tools - Visual Paradigm, Rational Software Architect. Visio, Argo UML, Rational Application Developer etc. platforms.

Course Code: BTCS507-18	Course Title: Computer Networks Lab	L:0;T:0; P:2	1Credits

Detailed List of Tasks:

Task1: To study the different types of Network cables and network topologies

Task2: Practically implement and test the cross-wired cable and straight through cable using clamping tool and network lab cable tester.

Task3: Study and familiarization with various network devices.

Task4: Familiarization with Packet Tracer Simulation tool/any other related tool. Task5: Study and

Implementation of IP Addressing Schemes

Task6: Creation of Simple Networking topologies using hubs and switches

Task7: Simulation of web traffic in Packet Tracer

Task8: Study and implementation of various router configuration commands

Task9: Creation of Networks using routers.

Task10: Configuring networks using the concept of subnetting

Task11: Practical implementation of basic network command and Network configuration commands like ping, ipconfig, netstat, tracert etc. for trouble shooting network related problems.

Task12: Configuration of networks using static and default routes.

Course Outcomes:

The students will be able to

1. Know about the various networking devices, tools and also understand the implementation of network topologies.
2. Create various networking cables and know how to test these cables.
3. Create and configure networks in packet tracer tool using various network devices and topologies.
4. Understand IP addressing and configure networks using the subnettin.
5. Configure routers using various router configuration commands.
6. Trouble shoot the networks by using various networking commands. Graphics Software’s.

Elective-I Lab

Course Code: BTCS511-18	**Course Title: Programming in Java Lab**	**L:0;T:0; P:2**	**1Credits**

To accomplish CO1;

1. WAP in Java to show implementation of classes.
2. WAP in Java to show implementation of inheritance.
3. WAP in Java to show Implementation of packages and interfaces.

To accomplish CO2;

4. WAP in Java to show Implementation of threads.
5. WAP in Java Using exception handling mechanisms.
6. WAP in Java to show Implementation of Applets.

To accomplish CO3;

7. WAP in Java to show Implementation of mouse events, and keyboard events.
8. WAP in Java to show Implementing basic file reading and writing methods.
9. Using basic networking features, WAP in Java

To accomplish CO4;

10. WAP in Java to show Connecting to Database using JDBC.

Project work: A desktop based application project should be designed and implemented in java.

Course Outcomes:

At the end of the course the student should be able to:

CO1. Implement the features of Java such as opeartors, classes, objects, inheritance, packages and exception handling

CO2. Design problems using latest features of Java like garbage collection, Console class, Network interface, APIs

CO3. Develop competence in Java through the use of multithreading, Applets etc

CO4. Apply advance concepts like socket and database connectivity, and develop project based on industry orientation.

Suggested Readings/Books

1. Herbert Schildt, The Complete Reference Java2, McGraw-Hill.
2. Deitel and Deitel, Java: How to Program, 6th Edition, Pearson Education.
3. James Edward Keogh, Jim Keogh, J2EE: The complete Reference, Mc Graw Hill

Course Code: BTCS 512-18	**Course Title: Web and Open Source Technologies Laboratory**	**L:0;T:0; P:2**	**1Credits**

Detailed List of Tasks:

1. Write an HTML page including javascript that takes a given set of integer numbers and shows them after sorting in descending order.
2. Write an HTML page that has one input, which can take multi-line text and a submit button. Once the user clicks the submit button, it should show the number of characters, words and lines in the text entered using an alert message. Words are separated with white space and lines are separated with new line character.
3. Write an HTML page that contains a selection box with a list of 5 countries. When the user selects a country, its capital should be printed next to the list. Add CSS to customize the properties of the font of the capital (color, bold and font size).
4. Create an XML document that contains 10 users information.
5. Using jQuery find all children in a specified class of a division
6. Find all elements of a form that are disabled
7. Create an input form and validate using jQuery. Highlight inputs elements if errors occur
8. Build a Single Page Application (SPA) using AngularJS.

Course Code: BTCS 513-18	Course Title: Programming in Python Laboratory	L:0;T:0; P:2	1Credits

Detailed List of Tasks:

1. Write a program to demonstrate different number data types in Python.
2. Write a program to perform different Arithmetic Operations on numbers in Python. 3. Write a program to create, concatenate and print a string and accessing sub-string from a given string.
3. Write a python script to print the current date in the following format “Sun May 29 02:26:23 IST 2017”
4. Write a program to create, append, and remove lists in python.
5. Write a program to demonstrate working with tuples in python.
6. Write a program to demonstrate working with dictionaries in python.
7. Write a python program to find largest of three numbers.
8. Write a Python program to convert temperatures to and from Celsius, Fahrenheit. [Formula: c/5 = f-32/9]

9. Write a Python program to construct the following pattern, using a nested for loop

```
 *

* *

* * *

* * * *

* * * * *

* * * *

* * *

* *

*
```

10. Write a Python script that prints prime numbers less than 20.
11. Write a python program to find factorial of a number using Recursion.
12. Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is a right triangle (Recall from the Pythagorean Theorem that in a right triangle, the square of one side equals the sum of the squares of the other two sides).
13. Write a python program to define a module to find Fibonacci Numbers and import the module to another program.
14. Write a python program to define a module and import a specific function in that module to another program.
15. Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.
16. Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.
17. Write a Python class to convert an integer to a roman numeral.
18. Write a Python class to implement pow(x, n)
19. Write a Python class to reverse a string word by word.

Elective-II Lab

Course Code: BTCS 517-18	Course Title:Mobile Application Development Lab	L:0;T:0; P:2	1Credits

Detailed List of Tasks:

1. Introduction to Android platform. Introduction to the tools used in the lab. Create a simple application
2. Understand the app idea and design user interface/wireframes of mobile app
3. Set up mobile app development environment
4. Write a program using activity class to show different events.
5. Write a program to convert text to speech.
6. Develop and debug mobile app components – User interface, services, notifications, broadcast receivers, data components
7. Using emulator to deploy and run mobile apps
8. Testing mobile app- unit testing, black box testing and test automation

Course Code: BTCS518-18	Course Title: Computer Graphics Lab	L:0;T:0; P:2	1Credits

Detailed List of Tasks:

1. WAP to draw different geometric structures using different functions.
2. Implement DDA line generating algorithm.
3. Implement Bresenham’s line generating algorithm.
4. Implement Mid-point circle line generating algorithm.
5. Implementation of Bresenham’s circle drawing algorithm.
6. Implementation of mid-point circle generating Algorithm.
7. Implementation of ellipse generating Algorithm.
8. WAP of color filling the polygon using Boundary fill and Flood fill algorithm.
9. To translate an object with translation parameters in X and Y directions.
10. To scale an object with scaling factors along X and Y directions.
11. Program of line clipping using Cohen-Sutherland algorithm.
12. To perform composite transformations of an object.
13. To perform the reflection of an object about major

Course Code: BTCS 519-18	Course Title: Internet of Things Laboratory Lab	L:0;T:0; P:2	1Credits

Detailed List of Tasks:

1. Familiarization with Arduino/Raspberry Pi and perform necessary software installation.
2. To interface LED/Buzzer with Arduino/Raspberry Pi and write a program to turn ON LED for 1 sec after every 2 seconds.
3. To interface Push button/Digital sensor (IR/LDR) with Arduino/Raspberry Pi and write a program to turn ON LED when push button is pressed or at sensor detection.
4. To interface DHT11 sensor with Arduino/Raspberry Pi and write a program to print temperature and humidity readings.
5. To interface motor using relay with Arduino/Raspberry Pi and write a program to turn ON motor when push button is pressed.
6. To interface OLED with Arduino/Raspberry Pi and write a program to print temperature and humidity readings on it.
7. To interface Bluetooth with Arduino/Raspberry Pi and write a program to send sensor data to smartphone using Bluetooth.
8. To interface Bluetooth with Arduino/Raspberry Pi and write a program to turn LED ON/OFF when ‘1’/’0’ is received from smartphone using Bluetooth.
9. Write a program on Arduino/Raspberry Pi to upload temperature and humidity data to things peak cloud.
10. Write a program on Arduino/Raspberry Pi to retrieve temperature and humidity data from thing speak cloud.
11. To install MySQL database on Raspberry Pi and perform basic SQL queries.
12. Write a program on Arduino/Raspberry Pi to publish temperature data to MQTT broker.