

GUJARAT TECHNOLOGICAL UNIVERSITY

COMPUTER ENGINEERING/INFORMATION TECHNOLOGY ARTIFICIAL INTELLIGENCE SUBJECT CODE: 2180703 SEMESTER: 8

Type of course: Regular

Prerequisite: Data Structures, Mathematics

Rationale: With the usage of Internet and World Wide Web increasing day by day, the field of AI and its techniques are being used in many areas which directly affect human life. Various techniques for encoding knowledge in computer systems such as Predicate Logic, Production rules, Semantic networks find application in real world problems. The fields of AI such as Game Playing, Natural Language Processing, and Connectionist Models are also important. Student should know some programming language for AI.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks						Total Marks
L	T	P	C	Theory Marks			Practical Marks			
				ESE (E)	PA (M)		ESE (V)		PA (I)	
					PA	ALA	ESE	OEP		
4	0	2	6	70	20	10	20	10	20	150

Content

Sr No	Course Contents	Teaching hours	Weightage
1	What is AI? : The AI Problems, The Underlying Assumption, What Is An AI Techniques, The Level Of The Model, Criteria For Success, Some General References, One Final Word.	2	4
2	Problems, State Space Search & Heuristic Search Techniques : Defining The Problems As A State Space Search, Production Systems, Production Characteristics, Production System Characteristics, And Issues In The Design Of Search Programs, Additional Problems. Generate-And-Test, Hill Climbing, Best-First Search, Problem Reduction, Constraint Satisfaction, Means-Ends Analysis.	5	10
3	Knowledge Representation Issues : Representations And Mappings, Approaches To Knowledge Representation.	3	5
4	Using Predicate Logic : Representation Simple Facts In Logic, Representing Instance And Isa Relationships, Computable Functions And Predicates, Resolution.	4	8

5	Representing Knowledge Using Rules : Procedural Versus Declarative Knowledge, Logic Programming, Forward Versus Backward Reasoning.	4	8
6	Symbolic Reasoning Under Uncertainty : Introduction To Non-monotonic Reasoning, Logics For Non-monotonic Reasoning.	4	8
7	Statistical Reasoning : Probability And Bays' Theorem, Certainty Factors And Rule-Base Systems, Bayesian Networks, Dempster-Shafer Theory, Fuzzy Logic.	3	5
8	Weak Slot-and-Filler Structures : Semantic Nets, Frames.	3	5
9	Strong Slot-and-Filler Structures : Conceptual Dependency, Scripts, CYC	3	5
10	Game Playing: Overview, And Example Domain : Overview, MiniMax, Alpha-Beta Cut-off, Refinements, Iterative deepening, The Blocks World, Components Of A Planning System, Goal Stack Planning, Nonlinear Planning Using Constraint Posting, Hierarchical Planning, Reactive Systems, Other Planning Techniques.	6	12
11	Understanding: What is understanding? , What makes it hard?, As constraint satisfaction	2	4
12	Natural Language Processing : Introduction, Syntactic Processing, Semantic Analysis, Semantic Analysis, Discourse And Pragmatic Processing, Spell Checking	5	8
13	Connectionist Models : Introduction: Hopfield Network, Learning In Neural Network, Application Of Neural Networks, Recurrent Networks, Distributed Representations, Connectionist AI And Symbolic AI.	4	6
14	Introduction to Prolog : Introduction To Prolog: Syntax and Numeric Function, Basic List Manipulation Functions In Prolog, Functions, Predicates and Conditional, Input, Output and Local Variables, Iteration and Recursion, Property Lists and Arrays, Miscellaneous Topics, LISP and Other AI Programming Languages.	8	12

Reference Books:

- 1 “Artificial Intelligence” -By Elaine Rich And Kevin Knight (2nd Edition) Tata Mcgraw-Hill
2. Artificial Intelligence: A Modern Approach, Stuart Russel, Peter Norvig, PHI
- 3 Introduction to Prolog Programming By Carl Townsend.
4. “PROLOG Programming For Artificial Intelligence” -By Ivan Bratko(Addison-Wesley)
5. “Programming with PROLOG” –By Klocksinn and Mellish.

Course Outcome:

After learning the course the students should be able to

- Understand various search methods
- Use various knowledge representation methods
- Understand various Game Playing techniques
- Use Prolog Programming language using predicate logic

List of Experiments:

1. Write a program to implement Tic-Tac-Toe game problem.
2. Write a program to implement BFS (for 8 puzzle problem or Water Jug problem or any AI search problem) .
3. Write a program to implement DFS (for 8 puzzle problem or Water Jug problem or any AI search problem)
4. Write a program to implement Single Player Game (Using Heuristic Function)
5. Write a program to Implement A* Algorithm.
6. Write a program to solve N-Queens problem using Prolog.
7. Write a program to solve 8 puzzle problem using Prolog.
8. Write a program to solve travelling salesman problem using Prolog.
9. Convert following Prolog predicates into Semantic Net

```
cat(tom).
cat(cat1).
mat(mat1).
sat_on(cat1,mat1).
bird(bird1).
caught(tom,bird1).
like(X,cream) :- cat(X).
mammal(X) :- cat(X).
has(X,fur) :- mammal(X).
animal(X) :- mammal(X).
animal(X) :- bird(X).
owns(john,tom).
is_coloured(tom,ginger).
```

10. Write the Conceptual Dependency for following statements.
(a) John gives Mary a book
(b) John gave Mary the book yesterday

Open Ended Problems:

1. Describe major subfields and paradigms of AI.
2. What are the major challenges in the field of AI?
3. How AI can be used to develop a better search Engine?

Major Equipments: Computer/Prolog Language

List of Open Source Software/learning website:

1. <http://www.journals.elsevier.com/artificial-intelligence/>
2. <https://www.technologyreview.com/s/534871/our-fear-of-artificial-intelligence/>
3. <http://www.sanfoundry.com/artificial-intelligence-mcqs-inductive-logic-unification-lifting-1/>

ACTIVE LEARNING ASSIGNMENTS: Preparation of power-point slides: which include videos, animations, pictures, graphics for better understanding theory and practical work. The faculty will allocate chapters/ parts of chapters to groups of students so that the entire syllabus can be covered. The power-point slides should be put up on the web-site of the College/ Institute, along with the names of the students of the group, the name of the faculty, Department and College on the first slide. The best three works should submit to GTU.

GUJARAT TECHNOLOGICAL UNIVERSITY
COMPUTER ENGINEERING/INFORMATION TECHNOLOGY
ANDROID PROGRAMMING
SUBJECT CODE:2180715
B.E. 8th SEMESTER

Type of course: Bachelor of Engineering

Prerequisite: Java programming and Object-oriented programming, Knowledge of RDBMS and OLTP

Rationale:

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks						Total Marks
L	T	P	C	Theory Marks			Practical Marks			
				ESE (E)	PA (M)		PA (V)		PA (I)	
					PA	ALA	ESE	OEP		
3	0	2	5	70	20	10	20	10	20	150

Content:

Sr. No.	Content	Total Hrs	% Weightage
1	<p>The Basics:</p> <ul style="list-style-type: none"> ● Hello World: Intro to Android, Why develop apps for Android?, Flavors of Android operating systems, Challenges of developing for Android (multiple OS, need backwards compatibility, need to consider performance and offline capability) ● Concept: Create Your First Android App: Overview of the development process - Java, Android Studio , Project layout in Android Studio, Target and minimum SDKs, Android Virtual Device (AVD) Monitor, Viewing logs in logcat and AVD, Android manifest file , App Architecture: An app consists of one or more activities. For an activity, write Java code and layout xml, and hook them together, and register the activity in the manifest file. ● Concept: Layouts, Views and Resources: Layout elements can be viewed and edited in Layout Editor and XML, Introduction to the range of UI elements, Resources (layouts, strings, styles, themes), Identifying 	10	

	<p>resources with IDs, Programmatically referencing resources using resource IDs, on Click attribute, Getting user input from a view, Programmatically changing UI elements, Layout Managers, Defining layouts for activities, inflating the layout.</p> <ul style="list-style-type: none"> ● Concept: Scrolling Views: How to make activities scrollable: compare ScrollView, ListView, RecyclerView , Getting the resource ID for a UI element by inflating a layout (needed for RecyclerView) , How to implement RecyclerView (requires layout managers and ViewHolders) , Performance implications of different kinds of scrolling UI elements ● Concept: Resources to Help You Learn: Resources to help you learn: Samples that ship with the SDK, Templates for projects, developer.android.com, Android developer blog , Android developer YouTube channel, Source code and samples in github, Stack overflow, Google search! ● Activities and Intents :About activities, Defining Activities , Activity Lifecycle , Activity navigation , About intents ,Explicit vs Implicit intents ,Passing info to new activity ,Returning data from activity ● The Activity Lifecycle and Managing State: Activity lifecycle , Activity lifecycle callback methods , Activity instance state ● Starting Activities with Implicit Intents: Starting activities by sending implicit intents, Intent filters and enabling your activities to receive intents, ShareCompat ● Testing and Debugging, and Backwards Compatibility: Debugging your apps, Testing your app, Support libraries 		
2	<p>User Interface:</p> <ul style="list-style-type: none"> ● User Input Controls: Getting user input , Changing keyboards , Buttons , Dialogs and pickers , Spinners, checkboxes, and radio buttons , Gestures , Speech recognition (not done), Sensors (not done) ● Menus: Options menu, contextual menus (floating and action bar), and popup menu, Adding menu items. Handling on Clicks from menus. ● Screen Navigation: Terminology, Different ways a user can navigate through an app, Action bar, Settings menu, Navigation drawer, Directed workflow (funnels), Best practices for navigation ● Themes and Styles: Best practices for themes and styles, Performance benefits for themes, When and how to use 	10	

	<p>drawables, best practices for drawable, When and how to use nine-patches, best practices for nine-patches, Tools for creating drawables</p> <ul style="list-style-type: none"> ● Material Design: What is material design? Material design best practices. Material Design guidelines, Implementing Material Design look and feel, with compatibility with previous versions, Support library for Material Design design, Transitions and Animations ● Adapt layouts for multiple devices and orientations: Why we need to consider different screen sizes and orientations , Screen density (dip or dp), How to create adaptive layouts using resources folders , Different ways to create images that scale nicely, Images and image formats and how they affect performance (download speeds). ● Accessibility: Why accessibility matters, Accessibility considerations: Color blindness, poor vision, poor hearing, physical limitations, Accessibility guidelines , Testing for accessibility , Screen readers, Making your app more accessible: Color and Contrast, button size --> Material Design guidelines, considerate layouts and navigation ● Localization: How to prep your app for localization, LTR and RTL (eg Arabic) text. ● Testing the User Interface: Automated testing of UIs, User testing your UI with real users, Using the Espresso and UI Automator frameworks for testing UIs 		
3	<p>Background Tasks:</p> <ul style="list-style-type: none"> ● Connect to the Internet: Background Tasks, Synchronous versus async tasks, What is the UI thread and when should you use it? , Example of a background task -- retrieving data over the internet, Creating background tasks. (schedule, send data, etc.) , Implementing AsyncTask (doInBackground(), callbacks) , Limitations of AsyncTask , Passing info to background tasks, Initiating background tasks, Scheduling background tasks (intro only, more later). ● Connecting to the Internet: Permissions, Building URIs, Opening and closing Internet connections, Parsing JSON in Android. (Because it's common.) , Sending requests and parsing response. ● AsyncTaskLoader: Intro to AsyncTaskLoader , loadInBackground() , AsyncTaskLoader callbacks , Benefits of loaders ● Broadcast Receivers: What is a Broadcast Receiver and a Broadcast Intent? , Broadcast Receiver Security and Lifecycle ● Services: What is a service? Long running task without a UI, Difference between Activity and Service , Start and 	10	

	<p>stop services, Lifecycle methods, Foreground services, IntentService class, App priority (critical, high, low), How to create a new Service.</p> <ul style="list-style-type: none"> ● Notifications: What is a Notification? , Notification Design Guidelines. ● Triggering, Scheduling, and Optimizing Background AlarmManager ● Transferring Data Efficiently: Less data, less often! Cell radio life cycle, Job Scheduler. Why to use Job Scheduler instead of SyncManager/SyncAdapter, Difference between alarms and job schedulers. 		
4	<ul style="list-style-type: none"> ● Data -- Saving, Retrieving, Loading ● Storing Data in your app: Internal versus external storage, Privacy, sharing, security, encryption of your data , Shared Preferences: Store private primitive data in key-value pairs , SQLite Databases: Store structured data in a private database , Store data on the web with your own network server, Firebase for storing and sharing data in the cloud, Concept: Preferences , What are Settings and Preferences? , Settings best practices (harder to take away settings than to add, for usability reasons, Storing and retrieving preferences as key/value pairs using SharedPreferences, Different Settings types, Settings menu, Using Activity and PreferenceFragments to allow users to set preferences ● Store data using SQLite database: Overview of SQLite,OpenHelper Android class , Querying (dev) Searching (user) databases , Best practices for using databases in Android , Best practices for testing your database ● Using Content Resolvers to access data: Content Providers and Content Resolvers work together, what is a content provider? , What is a content resolver? , How do they work together? , How to implement and use Content Resolvers ● Content Providers: When to implement content providers , How to implement content providers (overview), Content URIs , UriMatcher, Content Provider authorities , Required methods on ContentProvider (query, insert, delete, update) , MIME types , Contracts , Making content provider data accessible to other apps by modifying manifest, and protecting data with permissions. ● Using Loaders to Load and Display Data: Using loaders to asynchronously load data into an activity or fragment, Benefits of Loaders -- why use them? , Loader states (started, stopped, reset) , LoaderManager , Methods & callbacks to implement in Loaders: loadInBackground(), deliverResult() onStart/StopLoading(), 	10	

	onReset/Cancelled()), Registering listeners , Using CursorLoader with ContentProviders		
5	Polish and Publish <ul style="list-style-type: none"> ● Permissions: The permissions model ● Libraries: Using libraries ● Widgets: What are widgets? When to use them and how to implement them. ● Publishing your App: Different ways to monetize your app (overview only) ● Making and publishing APKs: Guidelines for publishing in Google Play , Make and sign the APK, Beta test your app , Publish your app to Google Play 	5	

Suggested Specification table with Marks (Theory):

Distribution of Theory Marks				
R Level	U Level	A Level	N Level	E Level

Legends: R : Remembrance ; U = Understanding; A = Application; N = Analyze; E = Evaluation and above Levels (Revised Bloom's Taxonomy)

Reference Books:

Course Outcome:

This course teaches final-year Computer Science students how to develop Android apps. To be able to understand the process of developing software for the mobile. To be able to create mobile applications on the Android Platform. To be able to create mobile applications involving data storage in SQLite database

List of Experiments:

1. Install Android Studio, Hello World, Logging

- Install Android Studio.
- Create a virtual device.
- Create and Run Hello World on emulator and device.
- Explore project layout.
- Generate and view log statements.
- Explore manifest file.

2. Practical: Make Your First Interactive UI

- Add Views and UI elements in Layout Editor to the app's home screen.
- Edit layout XML.
- Add click behavior to a button (show a toast).

- Change the UI through a button click.
- Write a method to use string resource to define a message to appear in the UI.
- Experiment with using different layouts.
- Explore other UI Elements in the Layout Manager.

3. Practical: Working with TextView Elements

- Use a scroll view for text with minor HTML formatting

4. Practical: Learning Resources

- Get answers from android.developer.com.
- Create new projects with different templates.
- Create a new project based on a sample in the SDK.
- Find out how to add a launcher icon for your app.
- Find out the most popular Android OS in India.

5. Practical: Create and Start Activities

- Create a new activity and layout
- Start the new activity from an existing activity with an explicit intent
- Pass user-entered information from one activity to the other
- Pass information back to the main activity

6. Practical: Lifecycle and State Callbacks

- Add Lifecycle callbacks
- Save and restore instance state

7. Practical: Start Activities with Implicit Intents

- Send an implicit intent to start an activity (open web site)
- Send an implicit intent to start an activity (open location)
- Use an intent filter to allow other apps to start an activity in your app
- Use `ShareCompat.IntentBuilder`

8. Practical: Using the Debugger

9. Practical: Testing your code

10. Practical: Use support library

11. Practical: Use Keyboards, Input Controls, Alerts, and Pickers

- Experiment in your app with different keyboards for user input, spelling suggestions, and auto-capitalization.
- Add a spinner input control for selecting one value out of a set of values.

Lecture hours:

- Create new app to show an alert, and record the user's selection (OK or Cancel).
MOVE TO CONCEPT.
- Update app to show date and time pickers and record the user's selections.

12. Practical: Use an Options Menu and Radio Buttons

- Set up an options menu and overflow menu
- Add items to the option (overflow) menu.
- Add radio buttons for user selection.
- Add Up navigation to the app bar.

13. Practical: Create a RecyclerView

- Create an activity that displays data in a RecyclerView.
- Make the items in the list clickable
- Add a floating action button to add items to the list

14. Practical: Theme, Custom Styles, Drawables

- Define and use a theme
- Define and use a custom style that uses a drawable

15. Practical: Add a FAB and Cards

- Create an app that uses a Floating Action Button (FAB)
- Add an activity that uses cards. Optionally, style the cards.
- Customize your app's theme and styles to use Material Design styles and colors.

16. Practical: Put yourself in the Users shoes

- Test your app for accessibility, using Talkback and Explore by Touch. Switch to monochrome color space
- Put in earplugs, can you still use your app?
- Wear the darkest glasses you can find, can you still use your gloves?
- Put on gloves, can you still use your app?
- How would you make one of the apps you have written so far more accessible?

17. Practical: Implement Localized Strings

- Create localized strings in your app
- Test by changing default language

18. Practical: Use Espresso to test your UI

- Use Espresso to Test Your UI

19. Practical: Create an AsyncTask

- Create a simple AsyncTask to do work in the background

20. Practical: Google APIs Explorer, JSON, Books API

- Use the Books API in the Google APIs Explorer to investigate request format and JSON response format
- Create a new app that uses the Books API and AsyncTask to search for the author of a book..
- Write the code to parse the response and extract and display the relevant information
- Debug errors when the Internet permission is missing
- Add the missing permission to the Android Manifest.
- Verify your fix by running and testing your app.

21. Practical: Use AsyncTaskLoader

- Use AsyncTaskLoader instead of AsyncTask to show book search results in a RecyclerView

22. Practical: BroadcastReceiver

- Create an app with a BroadcastReceiver

23. Practical: Notifications

- Trigger a Notification
- Add Actions to your Notification

24. Practical: Alarm Manager

- Implement an alarm manager

25. Practical: Job Scheduler

- Use JobScheduler to do background updates

26. Practical: Firebase Job Dispatcher

27. Practical: Get and Save User Preferences

- Implement Settings menu to allow users to enter preferences.
- Implement code to retrieve and user user preferences

28. Practical: Save user data in a database

- Create an app that allows users to enter notes
- Save the notes in a SQLite database

- Create an app that stores data in an SQL database.
- Display the data in a RecyclerView.
- Allow users to add, delete, and edit data items.

29. Practical: Querying and Searching a Database

30. Practical: Implement a Content Provider

- Add a content provider for your SQLite database

31. Practical: Use a ContentResolver to query your data

- Use a content resolver to query the database
- Display the results of the query
- Use the content resolver to add data to the database

32. Practical: Implement a Loader

- Implement a loader
- Register a Listener for the Loader
- Test the loader by checking that the Items in the UI update when the data generated by the loader changes
- Use an AsyncTaskLoader to update a scrolling list of notes titles as the user adds more notes
- Register a Listener for the Loader
- Test the loader by checking that the Items in the UI update when the underlying data changes

33. Practical: Beta testing your app

- Running a beta test on Google Play

GUJARAT TECHNOLOGICAL UNIVERSITY

COMPUTER ENGINEERING (07) & INFORMATION TECHNOLOGY (16) BIG DATA ANALYTICS SUBJECT CODE: 2180710 B.E. 8th SEMESTER

Type of course: Elective

Prerequisite: NA

Rationale: NA.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks						Total Marks
L	T	P	C	Theory Marks			Practical Marks			
				ESE (E)	PA (M)		ESE (V)		PA (I)	
					PA	ALA	ESE	OEP		
3	0	2	5	70	20	10	20	10	20	150

Content:

Sr. No.	Content	Total Hrs	% Weightage
1	INTRODUCTION TO BIG DATA Introduction– distributed file system–Big Data and its importance, Four Vs, Drivers for Big data, Big data analytics, Big data applications. Algorithms using map reduce	06	13
2	INTRODUCTION TO HADOOP AND HADOOP ARCHITECTURE Big Data – Apache Hadoop & Hadoop EcoSystem, Moving Data in and out of Hadoop – Understanding inputs and outputs of MapReduce -, Data Serialization.	12	25
3	HDFS, HIVE AND HIVEQL, HBASE HDFS-Overview, Installation and Shell, Java API; Hive Architecture and Installation, Comparison with Traditional Database, HiveQL Querying Data, Sorting And Aggregating, Map Reduce Scripts, Joins & Sub queries, HBase concepts, Advanced Usage, Schema Design, Advance Indexing, PIG, Zookeeper , how it helps in monitoring a cluster, HBase uses Zookeeper and how to Build Applications with Zookeeper.	08	15
4	SPARK Introduction to Data Analysis with Spark, Downloading Spark and Getting Started, Programming with RDDs, Machine	12	20

	Learning with MLlib.		
5	NoSQL What is it?, Where It is Used Types of NoSQL databases, Why NoSQL?, Advantages of NoSQL, Use of NoSQL in Industry, SQL vs NoSQL, NewSQL	05	12
6	Data Base for the Modern Web Introduction to MongoDB key features, Core Server tools, MongoDB through the JavaScript's Shell, Creating and Querying through Indexes, Document-Oriented, principles of schema design, Constructing queries on Databases, collections and Documents , MongoDB Query Language.	08	15

Suggested Specification table with Marks (Theory):

Distribution of Theory Marks					
R Level	U Level	A Level	N Level	E Level	C Level
10	20	25	28	16	0

Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create and above Levels (Revised Bloom's Taxonomy)

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Reference Books:

1. Boris lublinsky, Kevin t. Smith, AlexeyYakubovich, "Professional Hadoop Solutions", Wiley, ISBN: 9788126551071, 2015.
2. Chris Eaton,Dirk derooset al. , "Understanding Big data ", McGraw Hill, 2012.
3. BIG Data and Analytics , Sima Acharya, Subhashini Chhellappan, Willey
4. MongoDB in Action, Kyle Banker,Piter Bakkum , Shaun Verch, Dream tech Press
5. Tom White, "HADOOP: The definitive Guide", O Reilly 2012.
6. VigneshPrajapati, "Big Data Analyticswith R and Haoop", Packet Publishing 2013.
7. <http://www.bigdatauniversity.com/>
8. Learning Spark: Lightning-Fast Big Data Analysis Paperback by [Holden Karau](#)

Course Outcome:

Upon completion of this course, students will be able to do the following:

- Students will to build and maintain reliable, scalable, distributed systems with Apache Hadoop.
- Students will be able to write Map-Reduce based Applications

- Students will be able to design and build MongoDB based Big data Applications and learn MongoDB query language
- Students will learn difference between conventional SQL query language and NoSQL basic concepts
- Students will learn tips and tricks for Big Data use cases and solutions.

List of Experiments:

1. To understand the overall programming architecture using Map Reduce API
2. Store the basic information about students such as roll no, name, date of birth , and address of student using various collection types such as List, Set and Map
3. Basic CRUD operations in MongoDB
4. Retrieve various types of documents from students collection
5. To find documents from Students collection
6. Develop Map Reduce Work Application
7. Creating the HDFS tables and loading them in Hive and learn joining of tables in Hive

Design based Problems (DP)/Open Ended Problem:

1. Create a system which can use of Web search, web crawlers and web information retrieval.
2. Analyze and implement a system with Web graph mining.
3. Implement and Subscribe RSS News feeds to get latest news in India.

Major Equipment:

XMLSpy, RSS Feed, RSS Reader.

List of Open Source Software/learning website:

- <http://in.reuters.com/tools/rss>
- <http://www.altova.com/xmlspy.html>
- <https://www.w3.org/RDF/>

ACTIVE LEARNING ASSIGNMENTS: Preparation of power-point slides, which include videos, animations, pictures, graphics for better understanding theory and practical work – The faculty will allocate chapters/ parts of chapters to groups of students so that the entire syllabus to be covered. The power-point slides should be put up on the web-site of the College/ Institute, along with the names of the students of the group, the name of the faculty, Department and College on the first slide. The best three works should submit to GTU.

GUJARAT TECHNOLOGICAL UNIVERSITY

COMPUTER ENGINEERING CLOUD INFRASTRUCTURE AND SERVICES SUBJECT CODE: 2180712 B.E. 8th SEMESTER

Type of course: Bachelor of Engineering

Prerequisite: Nil

Rationale: The course presents a top-down view of cloud computing, from applications and administration to programming, infrastructure, billing and security. The topics include: overview of cloud computing, cloud systems, Load balancing in AWS, distributed storage systems, virtualization, security in AWS, and management services and Billing. Students will study state-of-the-art solutions for cloud computing developed by Amazon. Students will also apply what they learn in one programming assignments and one project executed over Amazon Web Services.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks						Total Marks
L	T	P	C	Theory Marks			Practical Marks			
				ESE (E)	PA (M)		ESE (V)		PA (I)	
					PA	ALA	ESE	OEP		
3	0	2	5	70	20	10	20	10	20	150

Content:

Sr. No.	Content	Total Hrs	% Weightage
1	Introduction to Cloud Technologies Introduction to the Cloud Computing, History of cloud computing, Cloud service options, Cloud Deployment models, Business concerns in the cloud.	03	10
2	Virtualization and Cloud Platforms Exploring virtualization, Load balancing, Hypervisors, Machine imaging, Cloud marketplace overview, Comparison of Cloud providers.	05	10
3	Introduction to AWS AWS history, AWS Infrastructure, AWS services, AWS ecosystem.	03	10
4	Programming, management console and storage on AWS Basic Understanding APIs - AWS programming interfaces, Web services, AWS URL naming, Matching interfaces and services, Elastic block store - Simple storage service, Glacier - Content delivery platforms.	06	15
5	AWS identity services, security and compliance Users, groups, and roles - Understanding credentials, Security policies, IAM abilities and limitations, AWS physical security - AWS compliance initiatives, Understanding public/private keys, Other AWS security capabilities.	04	10
6	AWS computing and marketplace Elastic cloud compute - Introduction to servers, Imaging computers,	04	10

	Auto scaling, Elastic load balancing, Cataloging the marketplace, AMIs, Selling on the marketplace.		
7	AWS networking and databases Virtual private clouds, Cloud models, Private DNS servers (Route 53), Relational database service – DynamoDB, ElastiCache, Redshift.	05	10
8	Other AWS services and management services Analytics services, Application services, Cloud security, CloudWatch, CloudFormation, CloudTrail, OpsWorks.	05	15
9	AWS billing and Dealing with disaster Managing costs, Utilization and tracking, Bottom line impact, Geographic and other concerns, Failure plans, Examining logs.	05	10

Suggested Specification table with Marks (Theory):

Distribution of Theory Marks				
R Level	U Level	A Level	N Level	E Level
10	18	18	18	06

Legends: R : Remembrance ; U = Understanding; A = Application; N = Analyze; E = Evaluation and above Levels (Revised Bloom's Taxonomy)

Reference Books:

- Cloud Computing Bible. Barrie Sosinsky. John Wiley & Sons. ISBN-13: 978-0470903568.
- Amazon Web Services For Dummies. Bernard Golden. For Dummies. ISBN-13: 978- 1118571835
- Rajkumar Buyya, Cloud Computing: Principles and Paradigms, John Wiley & Sons, First Edition
- Amazon Security overview whitepaper- <https://aws.amazon.com/whitepapers>
- IAM Getting started Guide <http://docs.aws.amazon.com/IAM/latest/UserGuide/getting-started.html>
- Amazon.com Mashups by Francis Shanahan, Wrox, Wiley Publishing Inc., ISBN-13: 978-0470097779, ISBN-10: 0470097779
- Amazon Web Services in Action by Michael Wittig and Andreas Wittig, Dreamtech Press, ISBN: 9789351198758
- Building Applications in the Cloud: Concepts, Patterns and Projects by Christopher M. Moyer, Pearson Addison-Wesley Professional, ISBN-10: 0321720202, ISBN-13: 978-0321720207
- Cloud Computing Design Patterns by Thomas Erl, Prentice Hall, ISBN-10: 0133858561, ISBN-13: 978-0133858563

Course Outcome:

After learning the course, the student will be able:

1. To explain the core concepts of the cloud computing paradigm: how and why this paradigm shift came about, the characteristics, advantages and challenges brought about by the various models and services in cloud computing.
2. To apply the fundamental concepts in datacenters to understand the tradeoffs in power, efficiency and cost by Load balancing approach.
3. To discuss system virtualization and outline its role in enabling the cloud computing system model.
4. To illustrate the fundamental concepts of cloud storage and demonstrate their use in storage systems such as Amazon S3 and HDFS.
5. To analyze various cloud programming models and apply them to solve problems on the cloud.
6. To understand various management and other distinguish services of AWS.
7. To analyze the billing of resources and other paradigm: how to deal with disasters.
8. To understand security and compliances for AWS.

9. To deploy applications over commercial cloud computing infrastructures such as Amazon

List of Assignments / Experiments:

1. Write pros and cons of Cloud Computing.
2. Summarize Cloud service models with real time examples.
3. Define Virtualization. Also list and explain different Hypervisors.
4. Discuss performance evaluation of service over cloud.
5. Software study on Hadoop, MapReduce and HDFS.
6. Create an AMI for Hadoop and implementing short [Hadoop](#) programs on the Amazon Web Services platform.
7. Create a scenario that use Amazon S3 as storage on cloud.

Design based Problems (DP)/Open Ended Problem:

1. Students will choose their project topic and work in teams of three or four to design, implement, and evaluate cloud applications using Hadoop on the Amazon Web Services platform.
2. Students will present, in groups of three or two, one research paper during the semester. These papers cover very recent developments in cloud computing. The presentations (using power point slides) will take place in class, and evaluation will be done based on active participation in discussions.

Major Equipment:

Computer system with latest hardware, High speed internet access, Java Environment with IDE (Eclipse or NetBeans), AWS Subscription

List of Open Source Software/learning website:

- CloudSim 3.0.3
- <http://www.cloudbus.org/>
- <https://aws.amazon.com/>
- <http://aws.amazon.com/documentation/>
- <http://docs.aws.amazon.com/IAM/latest/UserGuide/getting-started.html>

ACTIVE LEARNING ASSIGNMENTS: Preparation of power-point slides: which include videos, animations, pictures, graphics for better understanding theory and practical work. The faculty will allocate chapters/ parts of chapters to groups of students so that the entire syllabus can be covered. The power-point slides should be put up on the web-site of the College/ Institute, along with the names of the students of the group, the name of the faculty, Department and College on the first slide. The best three works should submit to GTU.

GUJARAT TECHNOLOGICAL UNIVERSITY

COMPUTER ENGINEERING/ INFORMATION TECHNOLOGY

SUBJECT NAME: iOS PROGRAMMING

SUBJECT CODE: 2180714

B.E. 8th SEMESTER

Type of course: Elective

Prerequisite: Basics of programming language, Concepts of OOP, DBMS, Server side scripting

Rationale: iOS Programming which is based on objective C is used to design & develop all kind of applications for the devices which runs on iOS operating system platform under X-CODE IDE.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks						Total Marks
L	T	P	C	Theory Marks			Practical Marks			
				ESE (E)	PA (M)		ESE (V)		PA (I)	
					PA	ALA	ESE	OEP		
3	0	2	5	70	20	10	20	10	20	150

Content:

Sr. No.	Content	Total Hrs	% Weightage
1	Fundamentals: Overview of MAC OS and X-CODE, Introduction to iPhone Architecture, Essential COCOA Touch Classes, Interface Builder, Nib File, COCOA and MVC Framework, Overview of features of latest iOS.	7	30%
2	Swift Basics: Basics of objective c, Need of transformation from objective c to swift, Data types, variables, constants, operators, Decision making statements, looping, arrays, dictionaries, functions, enumerations, structure, classes, inheritance	8	15%
3	iPhone application development: Auto Layout, Views, Outlets and Actions, Different View Controller: single view Controller, Master-Detail View Controller, Navigation View Controller, Managing Application Memory, Application delegate, Handling Keyboard Input, UI Controllers: Label, Button, Text Field, Slider, Switch, Progress View, Page Control, Table View, Collection View, Image View, Text View, Web View, Map View, Date Picker, Picker View, Search Bar, Gestures, push notification, Image Picker, QR Code Scanner, Audio and Video, Accelerometer, Location service, 3D touch, attribute tracking, Making the app live, overview of watchos.	17	35%
4	Database Management: SQLite, Web Services, JSON parsing, XML Parsing, alamofire	12	20%

Suggested Specification table with Marks (Theory):

Distribution of Theory Marks					
R Level	U Level	A Level	N Level	E Level	C Level
15	45	35	5*	-	-

Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create and above Levels (Revised Bloom's Taxonomy)

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Reference Books:

1. iOS 10 Programming Fundamentals with Swift by Matt Neuburg - O'Reilly Media Pub
2. Building iPhone and iPad Electronic Projects - Mike Westerfield - O'Reilly Media Pub.
3. Head First iPhone and iPad Development, 2nd Edition - Dan Pilone, Tracey Pilone - O'Reilly Media
4. Beginning iPhone and iPad Web Apps - Chris Apers, Daniel Paterson - Apress Pub

Course Outcome:

After learning the course the students should be able to:

1. Design iPhone and iPad application.
2. Develop iPhone and iPad application.
3. Upload iOS application on App Store

List of Experiments:

1. Installation of Xcode on MAC.
2. Write a program in Swift to check the number is even or odd.
3. Write a program in Swift to check the number is prime or not.
4. Write a program in Swift to check the string is palindrome or not.
5. Write a program in Swift that converts birth date into year, month and days.
6. Write a program to create a simple calculator in Swift.
7. Write a program to demonstrate different UI controllers.
8. Write an application to demonstrate the use of Table Control & Views.
9. Develop an iPhone application in which user can insert, update and delete the record in database.
10. Develop a program to generate a sign-up form which contains following fields: Username, Password, Gender, Birth-date, Country, Image, Submit, Terms and conditions. On successful registration attempt system must generate one alert message. (Label, Round rectangle button, Segmented control, Text field, Picker view, Data picker, Image view, Navigation)
11. Write an iPhone application which can play audio and video files.
12. Develop a medium size project using iOS programming with using all controllers, notifications, database & views.

Design based Problems (DP)/Open Ended Problem:

1. Create an iPhone application by which the user can put/take the things on rent
2. Create one music player for iPhone or iPad

Major Equipment:

- iMac/Macbook/machine which can run X-code.

List of Open Source Software/learning website:

- xcode
- <https://developer.apple.com>
- <http://www.tutorialspoint.com/swift/>

ACTIVE LEARNING ASSIGNMENTS:

Preparation of power-point slides, which include videos, animations, pictures, graphics for better understanding theory and practical work – The faculty will allocate chapters/ parts of chapters to groups of students so that the entire syllabus to be covered. The power-point slides should be put up on the web-site of the College/ Institute, along with the names of the students of the group, the name of the faculty, Department and College on the first slide. The best three works should submit to GTU.

GUJARAT TECHNOLOGICAL UNIVERSITY

COMPUTER ENGINEERING/INFORMATION TECHNOLOGY

SUBJECT NAME: IOT AND APPLICATIONS

SUBJECT CODE: 2180709

B.E. 8th SEMESTER

Type of course: Bachelor of Engineering

Prerequisite: Fundamentals of computer network, wireless sensor network, communication & internet technology, web technology, information security.

Teaching and Examination Scheme:

Teaching and Examination Scheme:										
Teaching Scheme			Credits	Examination Marks						Total Marks
L	T	P	C	Theory Marks			Practical Marks			
				ESE (E)	PA (M)		ESE (V)		PA (I)	
					PA	ALA	ESE	OEP		
3	0	2	5	70	20	10	20	10	20	150

Content

Sr. No.	Syllabus Content	No. of Hours
1	IoT & Web Technology The Internet of Things Today, Time for Convergence, Towards the IoT Universe, Internet of Things Vision, IoT Strategic Research and Innovation Directions, IoT Applications, Future Internet Technologies, Infrastructure, Networks and Communication, Processes, Data Management, Security, Privacy & Trust, Device Level Energy Issues, IoT Related Standardization, Recommendations on Research Topics.	8
2	M2M to IoT – A Basic Perspective – Introduction, Some Definitions, M2M Value Chains, IoT Value Chains, An emerging industrial structure for IoT, The international driven global value chain and global information monopolies. M2M to IoT-An Architectural Overview – Building an architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations.	10
3	IoT Architecture -State of the Art – Introduction, State of the art, Architecture Reference Model - Introduction, Reference Model and architecture, IoT reference Model, IoT Reference Architecture - Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views.	10
4	IoT Applications for Value Creations Introduction, IoT applications for industry: Future Factory Concepts, Brownfield IoT, Smart Objects, Smart Applications, Four Aspects in your Business to Master IoT, Value Creation from Big Data and Serialization, IoT for Retailing Industry, IoT For Oil and Gas Industry, Opinions on IoT Application and Value for Industry, Home Management, eHealth.	8

5	Internet of Things Privacy, Security and Governance Introduction, Overview of Governance, Privacy and Security Issues, Contribution from FP7 Projects, Security, Privacy and Trust in IoT-Data-Platforms for Smart Cities, First Steps Towards a Secure Platform, Smartie Approach. Data Aggregation for the IoT in Smart Cities, Security	8
---	--	---

Suggested Specification table with Marks (Theory):

Distribution of Theory Marks					
R Level	U Level	A Level	N Level	E Level	C Level
10	20	25	28	16	0

Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create and above Levels (Revised Bloom's Taxonomy)

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Reference Books:

1. Vijay Madiseti and Arshdeep Bahga, **“Internet of Things (A Hands-on-Approach)”**, 1st Edition, VPT, 2014
2. Francis daCosta, **“Rethinking the Internet of Things: A Scalable Approach to Connecting Everything”**, 1st Edition, Apress Publications, 2013
3. Cuno Pfister, **Getting Started with the Internet of Things**, O'Reilly Media, 2011, ISBN: 978-1-4493-9357-1

Course Outcome:

After learning the course, the student will be able:

1. Understand the vision of IoT from a global context.
2. Understand the application of IoT.
3. Determine the Market perspective of IoT.
4. Use of Devices, Gateways and Data Management in IoT.
5. Building state of the art architecture in IoT.
6. Application of IoT in Industrial and Commercial Building Automation and Real World Design Constraints.

List of Experiments:

1. Define and Explain Eclipse IoT Project.
2. List and summarize few Eclipse IoT Projects.
3. Sketch the architecture of IoT Toolkit and explain each entity in brief.
4. Demonstrate a smart object API gateway service reference implementation in IoT toolkit.

5. Write and explain working of an HTTP- to-CoAP semantic mapping proxy in IoT toolkit.
6. Describe gateway-as-a-service deployment in IoT toolkit.
7. Explain application framework and embedded software agents for IoT toolkit.
8. Explain working of Raspberry Pi.
9. Connect Raspberry Pi with your existing system components.
10. Give overview of Zetta.

Design based Problems (DP)/Open Ended Problem:

1. How do you connect and display your Raspberry Pi on a Monitor Or TV?
2. Create any circuitry project using Arduino.

Major Equipment:

Raspberry pi, Arduino

List of Open Source Software/learning website:

- <https://github.com/connectIOT/iottoolkit>
- <https://www.arduino.cc/>
- <http://www.zettajs.org/>
- Contiki (Open source IoT operating system)
- Arduino (open source IoT project)
- IoT Toolkit (smart object API gateway service reference implementation)
- Zetta (Based on Node.js, Zetta can create IoT servers that link to various devices and sensors)

ACTIVE LEARNING ASSIGNMENTS: Preparation of power-point slides, which include videos, animations, pictures, graphics for better understanding theory and practical work – The faculty will allocate chapters/ parts of chapters to groups of students so that the entire syllabus to be covered. The power-point slides should be put up on the web-site of the College/ Institute, along with the names of the students of the group, the name of the faculty, Department and College on the first slide. The best three works should submit to GTU.

GUJARAT TECHNOLOGICAL UNIVERSITY

COMPUTER ENGINEERING PYTHON PROGRAMMING SUBJECT CODE: 2180711 B.E. 8th SEMESTER

Type of course: Department Elective III

Prerequisite: Programming Concepts

Rationale: Python is a modern language useful for writing compact codes specifically for programming in the area of Server side Web development, Data Analytics, AI and scientific computing as well as production tools and game programming. This course covers the basics and advanced Python programming to harness its potential for modern computing requirements.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks						Total Marks
L	T	P	C	Theory Marks			Practical Marks			
				ESE (E)	PA (M)		ESE (V)		PA (I)	
					PA	ALA	ESE	OEP		
3	0	2	5	70	20	10	20	10	20	150

Content:

Sr. No.	Content	Total Hrs	% Weightage
1	Introduction to Python <ul style="list-style-type: none">• The basic elements of python• Branching Programs• Control Structures• Strings and Input• Iteration	4	7%
2	Functions, Scoping and Abstraction <ul style="list-style-type: none">• Functions and scoping• Specifications• Recursion• Global variables• Modules• Files• System Functions and Parameters	5	10%
3	Structured Types, Mutability and Higher-Order Functions <ul style="list-style-type: none">• Strings, Tuples, Lists and Dictionaries• Lists and Mutability• Functions as Objects	4	8%
4	Testing, Debugging, Exceptions and Assertions <ul style="list-style-type: none">• Types of testing – Black-box and Glass-box• Debugging• Handling Exceptions	4	7%

	<ul style="list-style-type: none"> • Assertions 		
5	Classes and Object-Oriented Programming <ul style="list-style-type: none"> • Abstract Data Types and Classes • Inheritance • Encapsulation and Information Hiding 	4	8%
6	Simple Algorithms and Data structures <ul style="list-style-type: none"> • Search Algorithms • Sorting Algorithms • Hash Tables 	5	10%
7	Advanced Topics I <ul style="list-style-type: none"> • Regular Expressions – REs and Python • Plotting using PyLab • Networking and Multithreaded Programming – Sockets, Threads and Processes, Chat Application 	10	20%
8	Advance Topics II <ul style="list-style-type: none"> • Security – Encryption and Decryption , Classical Cyphers • Graphics and GUI Programming – Drawing using Turtle, Tkinter and Python, Other GUIs 	12	30%

Suggested Specification table with Marks (Theory):

Distribution of Theory Marks				
R Level	U Level	A Level	N Level	E Level
10	18	18	18	06

Legends: R : Remembrance ; U = Understanding; A = Application; N = Analyze; E = Evaluation and above Levels (Revised Bloom's Taxonomy)

Reference Books:

1. John V Guttag. "Introduction to Computation and Programming Using Python", Prentice Hall of India
2. R. Nageswara Rao, "Core Python Programming", dreamtech
3. Wesley J. Chun. "Core Python Programming - Second Edition", Prentice Hall
4. Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, "Data Structures and Algorithms in Python", Wiley
5. Kenneth A. Lambert, "Fundamentals of Python – First Programs", CENGAGE Publication
6. Luke Sneeringer, "Professional Python", Wrox
7. "Hacking Secret Ciphers with Python", Al Sweigart, URL-
<https://inventwithpython.com/hacking/chapters>

Course Outcome:

After learning the course, the student will be able:

- To develop proficiency in creating based applications using the Python Programming Language.
- To be able to understand the various data structures available in Python programming language and apply them in solving computational problems.
- To be able to do testing and debugging of code written in Python.
- To be able to draw various kinds of plots using PyLab.
- To be able to do text filtering with regular expressions in Python

- To be able to create socket applications in Python
- To be able to create GUI applications in Python

Following guideline is to be kept in mind while framing the list:

- ✓ At least 25 programs are to be assigned.
- ✓ Programs should cover particular feature from syntactic concepts together with advanced topics and definition based on real life problem.
- ✓ Practical list should cover entire syllabus.
- ✓ List of Experiments :
 1. Develop programs to understand the control structures of python
 2. Develop programs to learn different types of structures (list, dictionary, tuples) in python
 3. Develop programs to learn concept of functions scoping, recursion and list mutability.
 4. Develop programs to understand working of exception handling and assertions.
 5. Develop programs for data structure algorithms using python – searching, sorting and hash tables.
 6. Develop programs to learn regular expressions using python.
 7. Develop chat room application using multithreading.
 8. Learn to plot different types of graphs using PyPlot.
 9. Implement classical ciphers using python.
 10. Draw graphics using Turtle.
 11. Develop programs to learn GUI programming using Tkinter.

Major Equipments:

- Latest PCs with related software

List of Open Source Software/learning website:

- Turtle - <https://docs.python.org/2/library/turtle.html>
- PyLab - <https://scipy.github.io/old-wiki/pages/PyLab>

ACTIVE LEARNING ASSIGNMENTS: Preparation of power-point slides: which include videos, animations, pictures, graphics for better understanding theory and practical work. The faculty will allocate chapters/ parts of chapters to groups of students so that the entire syllabus can be covered. The power-point slides should be put up on the web-site of the College/ Institute, along with the names of the students of the group, the name of the faculty, Department and College on the first slide. The best three works should submit to GTU.

GUJARAT TECHNOLOGICAL UNIVERSITY

COMPUTER ENGINEERING WEB DATA MANAGEMENT SUBJECT CODE:2180713 B.E. 8th SEMESTER

Type of course: Bachelor of Engineering

Prerequisite: Nil

Rationale: This course provides an in depth study of the area of web data management covering XML, XLink, and XPointer. The course primarily covers the state of the art in designing and building web applications and services, primarily focusing on issues and challenges that revolve around the management and processing of data. It also highlights the concepts of Ontology, RDF and OWL. It covers Building Web scale applications using web search, web crawlers, Web Graph mining, Map Reduce etc.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks						Total Marks
L	T	P	C	Theory Marks			Practical Marks			
				ESE (E)	PA (M)		ESE (V)		PA (I)	
					PA	ALA	ESE	OEP		
3	0	2	5	70	20	10	20	10	20	150

Content:

Sr. No.	Content	Total Hrs	% Weightage
1	Data Model Introduction to Modeling Web Data, Semistructured data, XML, Web Data Management with XML, XML Standards, XML and syntax, XML Data Model, XLink, and XPointer.	05	10
2	XPath and XQuery Introduction, Basics of XPath and XQuery, XPath: Steps and path expressions, Evaluation of path expressions, Generalities on axes and node tests, Axes, Node tests and abbreviations, Predicates, XPath 2.0; FLWOR expressions in XQuery: Defining variables - the for and let clauses, Filtering - the where clause, The return clause, Advanced features of XQuery; XPath foundations.	06	15
3	Typing Motivating Typing, Automata, Schema Languages for XML, Typing Graph Data: Graph Semistructured Data, Graph Bisimulation, Data guides.	04	10
4	XML Query Evaluation XML fragmentation, XML identifiers: Region-based identifiers, Dewey-based identifiers, Structural identifiers and updates; XML evaluation techniques: Structural join, Optimizing structural join queries, Holistic twig joins.	04	10
5	Ontologies, RDF, and OWL	05	15

	Introduction, Ontologies by example, Web resources, URI, namespaces, RDF, RDFS: RDF Schema, OWL, Ontologies and (Description) Logics.		
6	Querying Data through Ontologies Introduction, Querying RDF data: notation and semantics, Querying through RDFS ontologies, Answering queries through DL-LITE ontologies.	04	10
7	Data Integration Introduction, Containment of conjunctive queries, Global-as-view mediation, Local-as-view mediation, Ontology-based mediators, Peer-to-Peer Data Management Systems.	05	10
8	Building Web scale applications Web search, web crawlers, web information retrieval, Web graph mining and hot topics in web search, Distributed systems, failure management, Required properties of a distributed system, P2P networks, Hash-based structures, distributed indexing, Distributed computing with MapReduce.	07	20

Suggested Specification table with Marks (Theory):

Distribution of Theory Marks				
R Level	U Level	A Level	N Level	E Level
10	18	18	18	06

Legends: R : Remembrance ; U = Understanding; A = Application; N = Analyze; E = Evaluation and above Levels (Revised Bloom's Taxonomy)

Reference Books:

- Serge Abiteboul, Ioana Manolescu, Philippe Rigaux, Marie-Christine Rousset and Pierre Senellart, "Web Data Management", Cambridge University Press, 2011
- Bhavani Thuraisingham, "Web Data Management and Electronic Commerce", CRC Press, 2000
- Bhavani Thuraisingham, "XML Databases and the Semantic Web", CRC Press, 2002
- Athena Vakali and George Pallis, "Web Data Management Practices: Emerging Techniques and Technologies", IGI Publishing, 2007, ISBN-10: 1599042282; ISBN-13: 978-1599042282

Course Outcome:

After learning the course, the student will be able:

1. To understand the overall vision of the Semantic Web
2. To analyze the current technology stack (URIs, XML, RDF/S, OWL)
3. To understand how one could use these technologies for building something useful
4. To define and test an ontology
5. To define schema mappings
6. To install and use tools for semantic data management

List of Experiments:

1. Create an XML file defining an article in newspaper.
2. Create an XML file containing list of students. Also create stylesheet file to display list in an HTML format.
3. Create an XML file containing list of students. Using XPath display following information
 - Information of a student with ID No : 101
 - All the student in the sorted order according to their CGPA
4. Create an XForm to collect information from staff member regarding their publications. Details like Year of Publication, National/International, Title, Conference/Journal etc.
5. From the above gathered information, using XQuery find out the number of publication in a specific year.
6. Demonstrate the use of AJAX.
7. Study of XMLSPY tool.
8. Create an RSS for the events occurring in your institute.
9. Write a program to read the articles in RSS created in above practical.
10. Study of RDF (Resource Description Framework)

Design based Problems (DP)/Open Ended Problem:

1. Create a system which can use of Web search, web crawlers and web information retrieval.
2. Analyze and implement a system with Web graph mining.
3. Implement and Subscribe RSS News feeds to get latest news in India.

Major Equipment:

XMLSpy, RSS Feed, RSS Reader.

List of Open Source Software/learning website:

- <http://in.reuters.com/tools/rss>
- <http://www.altova.com/xmlspy.html>
- <https://www.w3.org/RDF/>

ACTIVE LEARNING ASSIGNMENTS: Preparation of power-point slides: which include videos, animations, pictures, graphics for better understanding theory and practical work. The faculty will allocate chapters/ parts of chapters to groups of students so that the entire syllabus can be covered. The power-point slides should be put up on the web-site of the College/ Institute, along with the names of the students of the group, the name of the faculty, Department and College on the first slide. The best three works should submit to GTU.