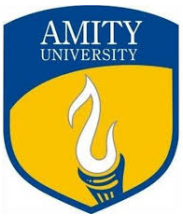
AMITY INSTITUTE OF INFORMATION TECHNOLOGY

Master of Computer Applications
MCA

FlexiLearn
-Freedom to Design your degree



AMITY UNIVERSITY MAHARASHTRA, MUMBAI

Programme Structure
Curriculum & Scheme of Examination
2020

First Year 2020-21
Second Year 2021-22

Sem 1	Sl. No.	Course Title	(L) Hours Per Week	(T) Hours Per Week	(P) Hours Per Week	Total Credits
	Core Courses					
	IFT4101	Operating Systems	3	0	0	3
	IFT4102	Android Programming	3	0	0	3
	IFT4103	Advanced Web Technologies (Node.js)	2	0	0	2
	IFT4104	Operating Systems Lab (Unix)	0	0	2	1
	IFT4105	Optimization Techniques	3	0	0	3
	IFT4106	Devops Concepts Lab	0	0	2	1
	IFT4107	Business Intelligence Lab (Sql Server SSIS)	0	0	2	1
	IFT4108	Python (Django & Kivy) Programming	2	0	0	2
	IFT4109	Java Lab	0	0	2	1
	IFT4110	Python Programming Lab	0	0	2	1
	IFT4111	Advanced Web Technologies (Node.js) Lab	0	0	2	1
	IFT4112	Android Programming Lab	0	0	2	1
	Value Enhancement Courses					
	CSS4151	Basics of Communication	1	0	0	1
	BEH4151	Self-Development and Interpersonal skills	1	0	0	1
	Open Elective					
		Foreign Language	3	0	0	3
					Total	25

Sem 2	Sl. No.	Course Title	(L) Hours Per Week	(T) Hours Per Week	(P) Hours Per Week	Total Credits
	Core Courses					
	1	Computer Networks	2	0	2	3
	2	Internet of Things	2	0	2	3
	3	Machine Learning	2	0	2	3
	4	Cloud Computing (Azure & AWS)	2	0	2	3
	5	Natural Languge Processing	2	0	2	3
	6	Data Structures and Algorithms	2	0	2	3
	7	C & C++ Lab	0	0	2	1
	Value Enhancement Courses					
	BEH4251	Behavioral Communication and Relationship Management	1	0	0	1
	CSS4251	Corporate Communication	1	0	0	1
	Open Elective					
	10	Foreign Language	3	0	0	3
	Concentration Elective for 3 Credits					
	11	Digital Marketing & SEO Optimization	2	0	2	3
		Software Engineering	2	0	2	3
		Advanced DevOps	2	0	2	3
		Data Visualization	2	0	2	3
		ASP & C# .NET	2	0	2	3
					Total	27

Sem 3	Sl. No.	Course Title	(L) Hours Per Week	(T) Hours Per Week	(P) Hours Per Week	Total Credits
	Core Courses					
	1	Image Processing	2	0	2	3
	2	Cyber Security	2	0	2	3
	3	Bigdata Processing	2	0	2	3
	4	Blockchain Architecture	2	0	2	3
	5	Parallel Processing	2	0	2	3
	6	Soft Computing	2	0	2	3
	7	UI/UX Design Lab	0	0	2	1
	8	Summer Internship Evaluation	0	0	0	5
	Value Enhancement Courses					
	CSS4351	Interpersonal Communication	1	0	0	1
	BEH 4351	Leading through Teams	1	0	0	1
	Open Elective					
	11	Foreign Language	3	0	0	3
	Concentration Elective for 3 Credits					
	12	Advanced Machine Learning	2	0	2	3
		Computer Architecture and Organization	3	0	0	3
		Advanced Technologies in Computer Science	2	0	2	3
		E-Commerce and ERP	2	0	2	3
		Software Project Management	2	0	2	3
					Total	32

Sem 4	Sl. No.	Course Title	(L) Hours Per Week	(T) Hours Per Week	(P) Hours Per Week	Total Credits
	Core Courses					
	1	Advanced Java: Spring	2	0	2	3
	2	Research Methodology	2	0	2	3
	3	Cyber Foreignsic	2	0	2	3
	4	Project / Dissertation	0	0	0	10
	Concentration Elective for 3 Credits					
	5	Advanced Tools for BigData Processing	2	0	2	3
		Software Testing and Quality Assurance	2	0	2	3
		Emerging Technologies in Computer Science	2	0	2	3
		Open Source Technologies (PHP & MySQL)	2	0	2	3
					Total	22

Course Code: IFT4101		Course Title: Operating Systems			
Total Credits	3	Theory	3	Tutorial	0

Module Number	Topics	No Hrs.
1	**Introduction** Operating system: Definition, Evolution and types of Operating Systems, Functions and Components or Structure of Operating Systems: Process management, memory management, Storage Management, Protection and Security, Special Purpose Systems, Computing Environment. System Structure: Services, System calls & Types, System programs	7
2	**Processes Management** Process concept, State model, Process scheduling, Threads CPU Scheduling**:** Job scheduling functions, Process scheduling, Scheduling Algorithms, Non Preemptive and preemptive Strategies, Algorithm Evaluation, Multiprocessor Scheduling. Inter-process Communication and Synchronization**:**Inter Process Communication,Principle of Concurrency, Producer Consumer Problem, Critical Section problem, Semaphores. Deadlock**:** System Deadlock Model, Deadlock Characterization, Methods for handling deadlock, Prevention strategies, Avoidance and Detection, Recovery from deadlock combined approach	15
3	**Memory Management** Memory Management Strategies: Contiguous Memory Allocation, Paging, Segmentation Virtual memory Management: Concept, Demand paging, Performance, Paged replacement algorithm, Allocation of frames, Thrashing, Cache memory, Swapping, Overlays.	10
4	**Device & Information Management** Principles of I/O hardware, Device controller, Device Drivers, Memory mapped I/O, Direct Access Memory, Interrupts, Interrupt Handlers, Buffering, Caching, Spooling,Disk organization, Disk space management, Disk allocation Method, Disk Scheduling, Disk storage. File Concept, Access Methods, Directory & Disk Structure, File System Structure, File System & Directory Implementation, Allocation Methods, Free Space Management	6
5	**The Unix System Case Study** History, Design Principle, Programmer Interface, User Interface, Process Management,Memory Managements, File management, Inter-process Communication.	6

Text & Reference Books	
1	Operating Systems Concepts, Silberschatz Galvin, Eighth Edition Addition Wesley Publication.
2	Modern Operating Systems, A S Tanenbaum, Prentice Hall of India New Delhi.
3	Operating Systems Internals & Design Principles , William Stallings ,Prentice Hall, Seventh Edition
4	Design of UNIX Operating System, Maurice J. Bauch, Prentice Hall of India.
Online Resources	
1	https://www.tutorialspoint.com/operating_system/os_quick_guide.htm

Course Code: IFT4102		Course Title: Android Programming			
Total Credits	3	Theory	3	Tutorial	0

Module Number	Topics	No Hrs.
1	**Module 1:**Android application development 1.1 Overview of Android 1.2 Devices running android 1.3 Why Develop for Android 1.4 Features of android 1.5 Architecture of Android, Libraries 1.6 Software development kit.	6
2	**Module 2:** Designing the user interface. 2.1 Introducing views , List of views and view groups 2.2 Introducing layouts, Creating new views, 2.3 Creating and using Menus	6
3	**Module 3:**Starting with Application Coding 3.1 Introducing Intents 3.2 Introducing Adapters 3.3 Using Internet Resources 3.4 Introducing Dialogs 3.5 Capturing Date and Time 3.6 Validating and Handling Input data. Accessing Location Based Services Application 4.1 Selecting Location Provider 4.2 Finding your location. 4.3 Creating map based activities. Data Storage, retrieval and Sharing 5.1 File system in android 5.2 Internal and external storage 5.3 Saving and loading files 5.4 File Management tools.	8
4	**Module 4:** Introduction to SQLite 6.1 Creating SQLite database, 6.2 Editing Tasks with SQLite 6.3 Cursors and content values 6.4 Working with Android database.	6
5	**Module 5:** Peer to peer to communication 7.1 Accessing Telephony Hardware 7.2 Introducing Android Instant Messaging 7.3 GTalkService : Using, binding & Making connection 7.4 Managing chat Sessions 7.5 Sending and receiving Data messages 7.6 Introducing SMS 7.7 Using, sending & receiving SMS Messages. Accessing Android Hardware 8.1 Audio, Video and Using the camera. 8.2 Introducing Sensor Manager 8.3 Android Telephony 8.4 Using Bluetooth 8.5 Manage network and Wi-Fi connections.	8
6	**Module 6:** Publishing Android Application to Market.	2

Text & Reference Books	
1	Professional Android™ Application Development Wrox Publications, Reto Meier
2	Hello Android, Introducing Google's Mobile Development Platform, Ed Burnette, Pragmatic Programmers,ISBN: 978-1-93435-617-3
3	Sams teach yourself Android application development, Lauren Dercy and ShandeConder, Sams publishing Reference Sites
4	
Online Resources	
1	https://developer.android.com

Course Code: IFT4103		Course Title: Advanced Web Technology			
Total Credits	3	Theory	2	Practical	1

Module Number	Topics	No Hrs.
1	Terminal: Basic commands of terminal, shell commands, ls, mkdir, rmdir, lsblk, mount, df, uname, ps, kill, service, batch, shutdown, touch, cat, head, tail, cp, mv, comm, less, In, cmp, dd, alias, call, history, find, grep, sed, clear, echo, sort, sudo, chmod, chown, man.	5
2	JavaScript: JS introduction, JS Output methods, scripting constructs, Data Types, Operators, Arithmetic, Assignment, functions, Objects, Arrays and its types, JS math, Get and Set, JS Looping Constructs, JS regular Expression, working form validations, JS Scope, working with error, this keyword, Form API, JS BOM, JS DOM, JS Objects, JS Functions	10
3	Node.JS: Introduction to Node JS, Node Application, building a RESTful API, Starting a server, parsing: request path, HTTP methods, Query String, header, payload, Server ping.	5
4	Building a Web APP GUI: Refactoring for GUI, using templates, serving Static Assets, AJAX Request, Index, Create Account, create session, edit, delete account, Dashboard, Building a CLI: Adding CLI, Handling event, commands: Exit, Man, Stats, List Users, More User Info, List checks, List Logs.	15
5	Advanced Node.JS: Gaining Stability, Creating Errors, using the debuggers, Linting with “strict”, Adding test runner, adding test unit, Adding API Test, Refactoring for performance, using performance hooks, using clusters, using child processes.	10

Text & Reference Books	
1	The Linux Command Line, 2nd Edition: A Complete Introduction
2	JavaScript and jQuery: Interactive Front-End Web Developmentby David McFarland
3	Beginning Node.js by Basarat Syed
4	Node.Js Web Development by David Herron
5	JavaScript: The Definitive Guide Book by David Flanagan
Online Resources	
1	https://nodejs.org/en/
2	https://nodejs.org/en/docs/
3	https://docs.microsoft.com/en-us/windows/terminal/command-line-arguments?tabs=windows
4	https://ubuntu.com/tutorials/command-line-for-beginners#1-overview
5	https://www.w3schools.com/nodejs/default.asp

Lab Course Code: IFT4104		Lab work Course Title: Operating System Lab (Unix)	
Total Credits	1	Practical : 2 Hrs per Week	Tutorial: 00

Assignment Number	Topics	No Hrs.
1	General Purpose Utility Commands	2
2	File System and Files Commands	2
3	Working with vi Editor	2
4	Shell Programming : Shell Program 1: Write a shell script to generate a multiplication table. a) Interactive version: The program should accept an integer n given by the user and should print the multiplication table of that n. b) Command line arguments version: The program should take the value of n from the arguments followed by the command. c) Redirection version: The value of n must be taken from a file using input redirection. Shell Program 2: Write a shell script that adds, subtracts, multiplies and divides the two given numbers. a) Interactive version: The program should accept two integersand the operation to be carried out from the user. Develop two versions of the program, one by using if and other using case construct. b) Command line arguments version: The program should supply the values of the integers from the command line arguments. Shell Program 3: Write a Shell Script which takes a command line argument of kms and by default converts that number into meters. Also provide options to convert km to dm and km tocm. Shell Program 4: Write a shell script to calculate sum of the digits of a three digit number. a) Interactive version: The program should accept the three digit numberfrom the user. b) Command line arguments version: The program should supply the three digit number from the command line arguments.	2
5	Shell Programming Shell Program 5: Write a Shell Script that performs and displays a count-down either from 10 (default) or from the value that is entered by theuser. Shell Program 6: Write a shell script that finds the value of one integer raised to the power of another. a) Interactive version: The program should accept two integers from the user. b) Command line arguments version: The program should supply the values of the	2

	integers from the command line arguments. Shell Program 7:Write a shell script to print first n terms of Fibonacci series a) Interactive version: The program should accept thenumber of terms to be printed from the user, interactively. b) Command line arguments version: The program should supply the values of the number of terms to be printed from the command line arguments.	
6	Shell Programming Shell Program 8: Write a shell script to print the GCD and LCM of two numbers a) Interactive version: The program should accept thetwo numbers from the user, interactively. b) Command line arguments version: The program should supply the two numbers from the command line arguments. Shell Program 9: Write a Shell Script that computes the factorial of a givennumber Shell Program 10: Write a Shell Script which creates the following menu and prompts for choice from user and runs the chosen command. i) Today's Date ii) Process of user iii) List of files iv) Quit UNIX	2
7	Shell Programming Shell Program 11: Write a Shell Script that works like a calendar reminding the user of certain things depending on the day of the week. Shell Program 12: Write a shell script to generate all combinations of 1, 2 and 3 using forloop. Shell Program 13:Write a shell script that prompts the user for the password. The user has maximum of 3 attempts. If the user enters the correct password, the message "Correct Password" is displayed else the message "Wrong Password" gets displayed.	2
8	Shell Programming Shell Program 14: Write a Shell Script using for loop, which displays the message "Welcome to the UNIX Lab". Shell Program 15: Write a Shell Script that receives two filenames as arguments. It should check whether content of the two files is same or not. If they are same, second file should bedeleted. Shell Program 16:Write a Shell Script that takes pattern and filename as command line arguments and displays the results appropriately i.e. pattern found/pattern not found.	2
9	Shell Programming Shell Program 17: Write a Shell Script that accepts a filename as a command line argument and finds out if it's a regular file or a directory. If it's a regular file, then performs various tests to see if it is readable, writeable and executable. Shell Program 18: Write a Shell Script that changes the extension of a group of files from txt to doc.	2

	Shell Program 19: Write a Shell Script which will redirect the output of the date command without the time into a file.	
10	Shell Programming Shell Program 20: Write a Shell Script to change the filename of all files in a directory from lower-case to upper- case. Shell Program 21: Write a Shell Script that examines each file in the current directory. Files whose names end in old are moved to a directory named old files and files whose names end in .c are moved to directory named cprograms. Shell Program 22: Write a shell script which reports names and sizes of all files in a directory (directory would be supplied as an argument to the shell script) whose size is exceeding 1000 bytes. The filenames should be printed in descending order of their sizes. The total number of such files should also be reported.	2
11	Shell Programming Shell Program 23: Write a shell script to identify all zero-byte files in the current directory and delete them. Before proceeding with deletion, the shell script should get a conformation from theuser. Shell Program 24: A shell script receives even number of filenames. Suppose four filenames are supplied, then the first file should get copied into second file, the third file should get copied into fourth and so on. If odd number of filenames is supplied then no copying should take place and an error message should be displayed. Shell Program 25: Write a Shell Script that accepts only three arguments from the command line. The first argument is the pattern string, the second argument is the filename in which the pattern is to be searches and the third argument is the filename in which the result is to be stored.	2
12	Shell Programming Shell Program 26: Write a shell script for renaming each file in the directory such that it will have the current shell PID as an extension. The shell script should ensure that the directories do not get renamed. Shell Program 27: Write a shell script that will receive any number of filenames as arguments. The shell script should check whether such files already exist. If they do, then it should be reported. The files that do not exist should be created in a sub-directory called mydir. The shell script should first check whether the sub-directory mydir exists in the current directory. If it doesn‟t exist, then it should be created. If mydir already exists, then it should be reported along with the number of files that are currently present inmydir.	2

Software & Tools Required	
1	Ubuntu 20.04

Text & Reference Books	
1	Unix Concepts & Applications, Sumitabha Das, Fourth Edition
2	UNIX Shell programming, Stephan G Kochan,Patrick Wood ,Third edition
Online Resources	
1	https://fog.ccsf.edu/~gboyd/cs160b/online/2-basics1/variables.html

Course Code: IFT4105		Course Title: Optimization Techniques			
Total Credits	3	Theory	3	Tutorial	0

Module Number	Topics	No Hrs.
1	Transportation problem and its mathematical formulation, northwest-corner method, least cost method and Vogel approximation method for determination of starting basic solution, algorithm for solving transportation problem,	9
2	Assignment problem and its mathematical formulation, Hungarian method for solving assignment problem.	9
3	Game theory: formulation of two person zero sum games, solving two person zero sum games, games with mixed strategies, graphical solution procedure.	6
4	Linear Programming Problems, Graphical Approach for Solving some Linear Programs. Convex Sets, Supporting and Separating Hyperplanes. Theory of simplex method, optimality and unboundedness, the simplex algorithm, simplex method in tableau format, introduction to artificial variables, two-phase method, Big-M method and their comparison. Duality, formulation of the dual problem, primal- dual relationships, economic interpretation of the dual, sensitivity analysis.	12

Text & Reference Books	
1	Optimization Techniques in Operations Research, C B Gupta, I. K. International Pvt Ltd, 2008
2	
3	
4	
Online Resources	
1	
2	
3	

Lab Course Code: IFT4106		Lab work Course Title: DevOps Lab	
Total Credits	1	Practical : 02 Hrs. Per Week	Tutorial : 00

Assignment Number	Topics	No Hrs.
1	Source Code Management using Git & GitHub, GitLab in collaborative teaming, Store, merge , version control of the code	4
2	Software and OS Configuration Management Tools, Managing Servers. Pushing the configurations using any one tool such as Puppet.	4
3	Azure DevOps Cloud Services for CI/CD.	4
4	Software and OS Configuration Management Tools, Managing Servers. Pushing the configurations using tool Ansible.	4
5	Lab on Continuous integration from different developers into a single piece of software using Jenkins.	4
6	Lab on Continuous integration from different developers into a single piece of software using Bamboo	4
	• Additional Assignments	
7	Continuous Testing using automation testing tools like Selenium, TestComplete, and TestingWhiz.	2
8	Software and OS Configuration Management Tools, Managing Servers. Pushing the configurations using tools Chef.	2
9	Continuous monitoring of the application performance Monitor the downtime, error logs etc. Tools : Negios, Zabbix, Splunk.	2
10	Containerization using Docker, Kubernetes and vagrant	2
	CI/CD using Scripts.	

Software & Tools Required	
1	Azure Cloud Services online account, Devops Accessibility
2	Jenkins, Chef, Puppet
3	Git, GitHub account
4	Dockers

Text & Reference Books	
1	Devops for Beginners, Joseph Joyner / Mihails Konoplovs, ISBN 9781682122105
2	Fundamentals of Software Engineering, Rajib Mall / PHI Learning, ISBN 978-9388028028
3	The DevOPS Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, by Gene Kim , Patrick Debois , et al.
4	Learning DevOps: The complete guide to accelerate collaboration with Jenkins, Kubernetes, Terraform and Azure DevOps, by Mikael Krief .
5	Effective DevOps with AWS: Implement continuous delivery and integration in the AWS environment, 2nd Edition by Yogesh Raheja , Giuseppe Borgese , et al.
Online Resources	

1	https://docs.microsoft.com/en-us/learn/modules/get-started-with-devops/3-what-is-azure-devops
2	https://docs.microsoft.com/en-us/learn/modules/assess-your-development-process/5-summary
3	
4	

Lab Course Code: IFT4107		Lab work Course Title: Business Intelligence Lab	
Total Credits	1	Practical : 2 Hrs per Week	Tutorial: 00

Assignment Number	Topics	No Hrs.
1	Introduction & Installation	2
2	Importing the legacy data from different sources and loading in the target system.	2
3	Creation of the Data staging area for the selected database and performing Extraction Transformation and Loading (ETL) process to construct the database in SQL server.	2
4	Creation of a data cube with suitable dimension tables and fact tables	2
5	Extracting data from the datawarehouse using MDX Query.	2
6	Visualization of Data using Pivot Table and Pivot Chart	2
7	Application of What –If analysis for data visualization	2
8	Implementation of Data Classification – Decision Tree	2
9	Implementation of Data Clustering	2
10	Implementation of Association Rule Mining	2
11	Implementation of Regression & Neural Network	2
12	Preparation of Report in SQL Server Reporting Services	2

Software & Tools Required	
1	SQL Server 2014
2	SQL Server Data Tools (SSDT) 2013
Text & Reference Books	
1	Efraim Turban, Ramesh Sharda, DursunDelen, “Decision Support and Business Intelligence Systems”, 9th Edition, Pearson 2013.
2	Ralph Kimball, Margy Ross, “The Data Warehouse Toolkit”, 3rd edition, Publisher: Wiley
3	Jiawei Han, MichelineKamber, Jian Pei “Data Mining: concepts and techniques”, 2nd Edition, Publisher: Elsevier/Morgan Kaufmann
4	William Inmon, “Building the Data Warehouse”, Wiley publication 4 th edition.
5	Randal Root, Caryn Mason,Pro SQL Server 2012 BI Solutions 1st Edition
Online Resources	
1	https://www.mssqltips.com/sqlservertutorial/9067/transformations-in-sql-server-integration-services-ssis/

Course Code: IFT4108		Course Title: Python (Django & Kivy) Programming			
Total Credits	4	Theory	2	Tutorial	0

Module Number	Topics	No Hrs.
1	**Python Introduction** Python Overview, Features, Installation & Environment setup, Basic syntax, Identifiers and Reserved words, Data Types, Types of Operators, Decision Making, Working with List and Tuples. Creating Dictionary: Declaring and adding elements, Accessing and deleting Dictionary, Python Lambda expression	5
2	**Functions and Libraries** Python Functions: Calling a function, Pass by reference vs value, Function Argument, The from...import Statement, Python file input-output. Python Classes and Objects, Class Inheritance, Regular Expression. Python Libraries: Matplotlib, Pandas, NumPy, BeautifulSoup, SciPy	4
3	**Django Introduction** Django and its Advantages, Feature, Django Web server, Django Architecture, MVC-MVT Pattern, Apps Life Cycle, Creating App in Django, Django Admin App.	5
4	**Django Forms and Template** Custom Field Validations in Django Models, Django ModelFormSet, Working with Hyperlinks, Page Redirection, Get and Post in Django. Django Template Language, Utilities of Templates, Creating Template Objects, Rendering Templates.	6
5	**Django Views** Creating views ,Django CRUD (Create, Retrieve, Update, Delete) Function Based Views, Class Based Generic Views Django (Create, Retrieve, Update,Delete)	6
6	**Kivy Introduction and Tools :** Introduction, Creating an Application,Kivy app life cycle,Kivy Architecture, Events and properties, Kivy Widgets,Button and Layout	4

Text & Reference Books	
1	Beginning Programming with Python ForDummiesLearning Python by Fabrizio Romano
2	Beginning Programming with Python for Dummies by John Paul Muller / Wiley India Pvt Ltd
3	Head First Python by Paul Barry / Shroff / O'Reilly Publisher
4	Mastering Django bu Nigel George
5	Django Web Development with Python by Samuel Dauzon,AidasBendoraitis, Arun Ravindran
6	Kivy Interactive Applications in Python by Roberto Ulloa
Online Resources	

Lab Course Code: IFT4109		Lab work Course Title: Java Programming Lab	
Total Credits	1	Practical : 02 Hrs. Per Week	Tutorial : 00

Assignment Number	Topics	No Hrs.
1	Programs on Basic Condition checking and Looping Constructs	4
	Programs on Basic Condition checking and Looping Constructs, reading data from the keyboard	
2	Exceptions – Checked, Unchecked, generics and enums.	4
	Java Keywords – Static, Final, volatile, synchronized transient, this, super etc. Program on Inheritance.	
3	Program on the abstract classes, Interfaces, Collections – List, Map, Set	4
	Programs on File Handling, File IO and Serialization	
4	Building GUI using AWT ,Swing & event handling.	4
5	Programs on Threads and multithreading.	4
6	Java Database connectivity programs.	2
7	Use of debugger to debug programs and profile an application.	4
	Understand when to use design patterns such as Singleton, Factory, Fly-weight, Builder, Object Pool, Iterator.	
	• Additional Assignments (Recommended).	
8	Java based web applications Using JSP / Servlets,	2
	Understand when to use design patterns such as Strategy, Visitor, Composite.	
9	Web Frameworks like Struts / Spring	2
	Service Oriented Architecture / Web Services – SOAP / REST	
10	Web Technologies like HTML, CSS, Javascript and JQuery	2
	Markup Languages like XML and JSON	
11	Frameworks like Swing, SWT, AWT, JavaFX (SWT only if you're building something on top of Eclipse)	2
	Understanding Java: RxJava	
12	Program on Applets.	2
	Write a simple client-server TCP service such as chat.	

Software & Tools Required	
1	Netbeans
2	Eclipse
3	JSDK

Text & Reference Books	
1	JAVA PROGRAMMING, BalaguruSamy / McGraw Hill Education, ISBN : 978-9351343202
2	Java - The Complete Reference, By Herbert Schildt
4	
Online Resources	
1	https://docs.oracle.com/en/java/

Lab Course Code: IFT4110		Lab work Course Title: Python Programming Lab	
Total Credits	2	Practical : 2	Tutorial : 0

Assignment Number	Topics	No Hrs.
1	Python Installation Programs on Data Types, Mathematical operators. Programs on Decision Making	2
2	Programs on List, Tuples and Dictionaries. Programs on Functions. Programs on OOPs concepts	4
3	Programs on Python Libraries: Matplotlib, Pandas, NumPy, BeautifulSoup, SciPy	5
4	Web Development Framework with Django: Installation of Django, setting up database Create Django project Creating an application, starting up the admin interface	5
5	Django Views: Creating Views and URL mapping Creating model and performing CRUD operations and linking models Use of generic views to create static page	5
6	Django Form and Template: ➢ Create Django form using get and post method for accepting username and password ➢ Create user facing application using Django template ➢ Use the Django-admin tool to create the project folder, basic file templates, and project management script (manage.py). • Use manage.py to create one or more applications. • Register the new applications to include them in the project. • Hook up the url/path mapper for each application	5
7	Designing GUI with Kivy ➢ Installation of Kivy and running Hello world program ➢ Create a page layout using kivy.kv file ➢ Create Kivy application for addition, subtraction, multiplication and division. ➢ Creating Desktop and Mobile applications	4

Software & Tools Required	
1	Anaconda
2	Idle Python
3	Django
4	Kivy

Text & Reference Books	
1	Python Programming Fundamentals - A Beginner's Handbook by Nishay Kumar Hegde
2	A Byte of Python C.H. Swaroop
3	
Online Resources	
1	https://www.python.org/
2	

Lab Course Code: IFT4111		Lab work Course Title: Advanced Web Technology Lab	
Total Credits	1	Practical : 02 Hrs. per Week	Tutorial : 00

Assignment Number	Topics	No Hrs.
1	Starting a server service using node.js	2
2	Parsing Request path and HTTP methods using node.js	2
3	Parsing Query string and header using Node.js	2
4	Parsing Payload and Routing Request and returning to JSON	2
5	Logging to file and the console in background using Node.js	2
6	Using Node.js making AJAX requests.	2
7	Creating account and session, deleting session.	2
8	Create a Dashboard using node.js	2
9	Working with CLI Commands (Part 1)	2
10	Working with CLI Commands (Part 2)	2
11	Working with CLI command (Part 3)	2
12	Working with stability of the transaction system using node.js	2

Software & Tools Required	
1	Terminal (Linux, windows, MacOS): sudo privileges
2	Node.js 8.x.x LTS
3	Text Editor : Atom, Sublime
4	

Text & Reference Books	
1	The Linux Command Line, 2nd Edition: A Complete Introduction
2	JavaScript and jQuery: Interactive Front-End Web Developmentby David McFarland
3	Beginning Node.js by Basarat Syed
4	Node.Js Web Development by David Herron
Online Resources	
1	Github.com for examples
2	https://atom.io/
3	https://nodejs.org/en/
4	https://www.w3schools.com/nodejs/default.asp

Lab Course Code: IFT4112		Lab work Course Title: Android Programming Lab	
Total Credits	1	Practical : 02 Hrs. Per Week	Tutorial : 00
Objectives		Student to enhance their Android Programming Skills. Android concepts such as Views and view groups, Layouts, Creating Menus Intents, Adapters, Dialogs, location based services, file handlings, CRUD operation on SQlite, Gtalk, Audio, Video can be included.	

Assignment Number	Topics	No Hrs.
1	Write a program to print Hello World on the First Activity Android	2
	Write program to show the addition of the two numbers on the same activity.	
2	Write a program to display the information of the first activity to the second activity.	2
3	Perform the local database SQLite and demonstrate on the Activity for the daily expenses storage in your android app.	2
4	Perform a setup of your Real Android Device to execute and debug .apk.	2
5	Write Android app to perform the database connectivity to the Web-Server. (Apache web server.)	2
6	Write a program for the calculator in Android app.	2
7	Write a program to vibrate the phone for 5 seconds.	2
8	Write a program to enable Bluetooth and Wi-fi through the radio button.	2
9	Write android app for calling to mobile number.	2
10	Create an Android App to run audio file.	2
11	Write a program to run video file.	2

Software & Tools Required	
1	Android Studio 4.0

Text & Reference Books	
1	Android Application Development (With Kitkat Support), Black Book, by Pradeep Kothari and Kogent Learning Solutions Inc.
2	Android Programming Unleashed, 1e, by Harwani.
3	Android Programming for Beginners - Second Edition, by John Horton
4	
Online Resources	
1	https://developer.android.com/courses