5.1 INDUSTRIAL TRAINING - II

L	P
-	2

RATIONALE

Industrial training will help the students to understand the working environmentof relevant industries. The student will learn to work in team to solve the industrial problems. It will also give exposure about the present and future requirements of the relevant industries. This trainingis very important for development of required competencies and skills for employment and start– ups.

COURSE OUTCOMES

After undergoing the training, the students will be able to:

CO1: Understand the working environment of industries
CO2: Take necessary safety precautions and measures.
CO3: Learn about present and future requirement of industries.
CO4: Work in team for solving industrial problems.
CO5: Develop competencies and skills required by relevant industries.
CO6: Develop writing, speaking and presentations skills.

PRACTICAL EXERCISES

1. Report writing based on industrial training.
2. Preparation of Power Point Slides based on industrial training and presentation by the candidate.
3. Internal Evaluation based on quality of Report, PPT preparation, PPT presentation and answer to queries.
4. External Evaluation based on quality of Report, PPT preparation, PPT presentation and answer to queries.

GUIDELINES

Students will be evaluated based on Industrial training report and their presentation using Power Point about the knowledge and skills gained during the training. The Headof the Department will depute faculty coordinators by assigning a group of students to each. The coordinators will mentor and guide the students in preparing the PPTs for final presentation. The following performance parameters are to be considered for assessment of the students out of 100 marks:

	Parameter	**Weightage**
i	Industrial assessment of the candidate by the trainer	40%
ii	Report Writing	20%
iii	Power Point Presentation	20%
iv	Viva-voce	20%

5.2 WEB TECHNOLOGIES

L	P
3	4

RATIONALE

A computer engineering diploma student should have good exposure of various web technologies. This course will develop competency amongst the students to design professional database backed dynamic and feature based web sites. The course covers the use of programming with PHP and the concepts of database using MySQL.

COURSE OUTCOMES

After undergoing the subject, the students will be able to:

CO1: Develop different portal using HTML.
CO2: Perform various logical operations in PHP.
CO3: Create database using MySQL.
CO4: Install and configure Joomla.
CO4: Perform database connectivity using PHP.

DETAILED CONTENTS

UNIT I
DEVELOPING PORTALS USING HTML

Introduction to HTML 5 and CSS 3. Basic structure of HTML, designing a web page, inserting links images, horizontal rules, comments. Formatting text, title, headings, colors, fonts, sizes, simple tables and forms. HTML tags, hyperlinks. Adding graphics and images, image maps, image files. Using tables, forms, style sheets and frames. Floating of web site/pages.

UNIT II
PHP

Introduction to PHP: How PHP Works , The php.ini File, Basic PHP Syntax, PHP variables, statements, operators, decision making, loops, arrays, strings, forms, get and post methods, functions.

Introduction to cookies, storage of cookies at client side, Using information of cookies. Creating single or multiple server side sessions. Timeout in sessions, Event management in PHP. Introduction to content management systems based on PHP.

UNIT III

MySQL

Introduction to MySQL, connecting to MySQL, database, creation, insertion, deletion and retrieval of MySQL data using PHP.

UNIT IV

JOOMLA BASICS AND ADMIN

Installing Wamp Server -Installing Joomla on Web Server, Joomla global configuration -Article manager -Archive manager-FrontPage manager -Section manager - Category manager- Media Manager-Menu manager -Component manager -Content Manager-Extensions manager-Module manager-Plugin manager-Template manager-How to install a new module-How to install a new template-How to install a new plugin-How to install a new component-Understanding the concept of Joomla positions -Changing the layout structure by changing the module position.

UNIT V

JOOMLA FRONTEND

Understanding Basic Joomla Template-Customizing Joomla template-Building Custom Joomla Template-Understanding Templatedetails.xml File-Creating Templatedetails.xml File using tmpl_builder Linking CSS-Linking JavaScript-Understanding Include-Displaying Content in Xhtml-Creating Template installation Package-Creating Custom Forms-Changing the Form Appearance using CSS.

PRACTICAL EXERCISES

1. Design PHP based web pages using correct PHP, CSS, and XHTML syntax, structure.
2. Create Web forms and pages that properly use HTTP GET and POST protocol as appropriate.

3 Design SQL language within MySQL and PHP to access and manipulate databases.

4. Install and configure both PHP and MySQL.

5 Create PHP code that utilizes the commonly used API library functions built in to PHP.

6. Design and create a complete web site that demonstrates good PHP/MySQL client/server design.

7. To store a cookie using PHP on client side.
8. To save the user session on server side.
9. Design website using Joomla.

RECOMMENDED BOOKS

1. Sams Teach Yourself PHP, MySQL, and Apache All in One" by Julie C.
2. Meloni, Publisher: SAMS ,ISBN 0-672-32976-X.
3. Web enabled development application by Ivan Byross: Commercial; TMH.
4. HTML, CSS, JavaScript, Perl, Python and PHP by Schafer Textbooks; Wiley India.
5. E-books/e-tools/relevant software to be used as recommended by AICTE/HSBTE/NITTTR.

RECOMMENDED WEBSITES

1. http://swayam.gov.in

INSTRUCTIONAL STRATEGY

This is hands on practice based subject and topics taught in the class should be practiced in the Lab regularly for development of required skills among the students. This subject contains five units of equal weightage.

5.3 PYTHON PROGRAMMING

L	P
3	4

RATIONALE

This subject introduces to the students the Python language. Upon completion of this subject, the student will be able to write non trivial Python programs dealing with a wide variety of subject matter domains. Topics include language components, the IDLE/IDE environment, control flow constructs, strings, I/O, collections, classes, modules, and regular expressions.

COURSE OUTCOMES

After undergoing the subject, the students will be able to:

CO1: Implement Python programs utilizing arithmetic expressions, repetition, file Inputand Output.
CO1: Demonstrate the use of the built-in data structures in Python.
CO3: Employ control structures, functions, and arrays to create Python programs.
CO4: Understand the concepts of object-oriented programming as used in Python.
CO5: Define the use of GUI and databases using Python.

DETAILED CONTENTS

UNIT I

The way of the program: The Python programming language, What is a program? What is debugging?, Syntax errors, Runtime errors, Semantic errors, Experimental debugging.
Variables, Expressions and Statements: Values and data types, Variables, Variable names and keywords, Statements, Evaluating expressions, Operators and operands, Type converter functions, Order of operations, Operations on strings, Input, Composition, The modulus operator.
Conditionals: Boolean values and expressions, Logical operators, Simplifying Boolean Expressions, Conditional execution, Chained conditionals, Nested conditionals, The return statement, Logical opposites.

UNIT II

Iteration: Assignment, Updating variables, The for loop, The while statement, The Collatz 3n + 1 sequence, Tables, Two-dimensional tables, Paired Data, Nested Loops for Nested Data.

Strings: Working with strings as single things, Working with the parts of a string, Length, Traversal and the for loop, Slices, String comparison, Strings are immutable, The in and not in operators, A find function, Looping and counting, Optional parameters, The built-in find method,The split method, Cleaning up your strings, The string format method.

Tuples: Tuples are used for grouping data, Tuple assignment, Tuples as return values, Composability of Data Structures.

Lists: List values, Accessing elements, List length, List membership, List operations, List slices, Lists are mutable, List deletion, Objects and references, Aliasing, Cloning lists, Lists and for loops, List parameters, List methods, Pure functions and modifiers, Functions that produce lists, Strings and lists, list and range, Nested lists, Matrices.

Functions: Functions with arguments and return values.

UNIT III

Modules: Random numbers, The time module, The math module, Creating your own modules, Namespaces, Scope and lookup rules, Attributes and the dot operator.

Files: About files, Writing our first file, Reading a file line-at-a-time, Turning a file into a list of lines, Reading the whole file at once, Working with binary files, Directories, fetching something from the web. **List Algorithms:** Linear search, Binary search, Merging two sorted lists.

UNIT IV

Object oriented programming: Classes and Objects- The Basics, Attributes, Adding methods to our class, Instances as arguments and parameters, Converting an instance to a string, Instances as return values, Objects are mutable, Sameness, Copying.

Exceptions: Catching exceptions, raising our own exceptions, the finally clause of the try statement

Inheritance: Polymorphism, Generalization, Pure functions.

UNIT V

GUI: Creating Graphical User Interfaces, Using Module Tkinter, Building a Basic GUI, Models, Views, and Controllers, Customizing the Visual Style, Few More Widgets.

Databases: Overview, Creating and Populating, Retrieving Data, Updating and Deleting, Using NULL for Missing Data, Using Joins to Combine Tables, Keys and Constraints, Advanced Features.

PRACTICAL EXERCISES

Part A

1. Let list1 and list2 be two lists of integers. Implement function sublist() that takes as input lists list1 and list2 and returns True if list1 is a sublist of list2, and False otherwise.

   ```
   >>> sublist([15, 1, 100], [20, 15, 30, 50, 1, 100])
   True
   >>> sublist([15, 50, 20], [20, 15, 30, 50, 1, 100])
   False
   ```

2. Write function vowelCount() that takes a string as input and counts and prints the numberof occurrences of vowels in the string.

   ```
   >>> vowelCount('Le Tour de France')
   a, e, i, o, and u appear, respectively, 1, 3, 0, 1, 1 times.
   ```

3. The cryptography function crypto() takes as input a string (i.e., the name of a file in the current directory). The function should print the file on the screen with this modification: Every occurrence of string 'secret' in the file should be replaced with string 'xxxxxx'.

   ```
   >>> crypto('crypto.txt')
   I will tell you my xxxxxx. But first, I have to explainwhy
   it is a xxxxxx.
   And that is all I will tell you about my xxxxxx.
   ```

4. Write a function stats() that takes one input argument: the name of a text file. The function should print, on the screen, the number of lines, words, and characters in the file;your function should open the file only once.

   ```
   >>>stats('example.txt')
   line count: 3
   word count: 20
   character count: 98
   ```

5. Implement function distribution () that takes as input the name of a file (as a string). This one-line file will contain letter grades separated by blanks. Your function should print the distribution of grades, as shown.

   ```
   >>> distribution('grades.txt')6
   students got A
   2 students got A-3
   ```

students got B+2
students got B 2
students got B-4
students got C 1
student got C- 2
students got F

6. The function censor () takes the name of a file (a string) as input. The function should open the file, read it, and then write it into file censored.txt with this modification: Every occurrence of a four-letter word in the file should be replaced with string 'xxxx'.

```
>>> censor ('example.txt')
```

Note that this function produces no output, but it does create file censored.txt in the current folder.

7. Create a dictionary for phones and their prices. Write functions to add a new entry (phone:price) ,search for a particular phone and retrieve it's price, given price findphones with same price , remove an entry, display all phones sorted according to price. [Program must be menu driven]

8. Write a Python program that prompts the user to enter a list of first names and stores them in a list. The program should display how many times the letter 'a' appears within the list.

9. Write a Python program that prompts the user to enter integer values for each of two lists.It then should displays whether the lists are of the same length, whether the elements in each list sum to the same value, and whether there are any values that occur in both lists.

10. Implement and test a Python program that determines if all parentheses in an entered line of code form matching pairs. Note: Pairs of parentheses may be nested.

11. Suppose variable s has been assigned in this way:

s = '''It was the best of times, it was the worst of times; it

was the age of wisdom, it was the age of foolishness; it was the
epoch of belief, it was the epoch of incredulity; it was ...''' Then
do the following, in order, each time:

(a) Write a sequence of statements that produce a copy of s, named newS, in whichcharacters ., ,, ;, and \n have been replaced by blank spaces.

(b) Remove leading and trailing blank spaces in newS (and name the new string newS).

(c) Make the all characters in newS lowercase (and name the new string newS).

(d) Compute the number of occurrences in newS of string 'it was'.

(e) Change every occurrence of was to is (and name the new string newS).

(f) Split newS into a list of words and name the list listS.

12. The function avgavg() takes as input a list whose items are lists of three numbers. Each three-number list represents the three grades a particular student received for a course. For example, here is an input list for a class of four students:
[[95,92,86], [66,75,54],[89, 72,100],[34,0,0]]

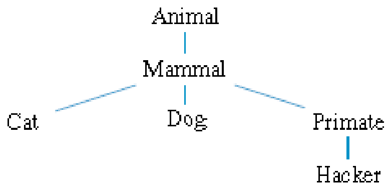
The function avgavg() should print, on the screen, two lines. The first line will contain a list containing every student's average grade. The second line will contain just one number: the average class grade, defined as the average of all student average grades.

```
>>> avgavg([[95, 92, 86], [66, 75, 54],[89, 72, 100], [34, 0, 0]])

[91.0, 65.0, 87.0, 11.333333333333334]

63.5833333333
```

13. Implement function names () that takes no input and repeatedly asks the user to enter the first name of a student in a class. When the user enters the empty string, the function should print for every name the number of students with that name.

```
>>> names ()

Enter next name: Valerie
Enter next name: Bob Enter
next name: ValerieEnter
next name: AmeliaEnter
next name: Bob Enter next
name:
There is 1 student named Amelia
There are 2 students named Bob
There are 2 students named Valerie
```

14. Consider the class tree hierarchy:



Implement six classes to model this taxonomy with Python inheritance. In class Animal, implement method speak() that will be inherited by the descendant classes of Animal as is.

Complete the implementation of the six classes so they exhibit this behavior:

```
>>> garfield = Cat()
>>> garfield.speak()
Meeow
>>> dude = Hacker()
>>> dude.speak( )
Hello world!
```

Part B

1. Numerologists claim to be able to determine a person's character traits based on the numeric value” of a name. The value of a name is determined by summing up the values of the letters of the name where 'a' is 1, 'b' is 2, 'c' is 3 etc., up to 'z' being 26. For example,the name "“Zelle"”would have the value 26+5+12+12+5=60 (which happens to be a very auspicious number, by the way). **Write a program that calculates the numeric value of a single name provided as input.** (Hint: Use dictionary, strings and its methods)
2. Expand your solution to the previous problem to allow the **calculation of a complete name** such as “John Marvin Zelle”” or “”John Jacob Jingleheimer Smith”. The total value is just the sum of the numeric values of all the names.
3. **Write a python program with function inner_product(x,y)** that computes the inner product of two (same length) lists. For example: list1=[1,2,3,4,5] and list2=[1,2,3,4,5]. The inner product list is inner_product=[1,4,9,16,25].
4. The **Sieve of Eratosthenes** is an elegant **algorithm** for finding all of the prime numbers up to some limit n. The basic idea is to first create a list of numbers from 2 to n. The first number is removed from the list, and announced as a prime number, and all multiples of this number up to n are removed from the list. This process continues until the list is empty.
   a) For example, if we wished to find all the primes up to 10, the list would originally contain 2, 3, 4, 5, 6, 7, 8, 9, 10.
   b) The 2 is removed and announced to be prime.
   c) Then 4, 6, 8, and 10 are removed, since they are multiples of 2.
   d) That leaves 3, 5, 7, 9.
   e) Repeating the process, 3 is announced as prime and removed, and 9 isremoved because it is a multiple of 3.
   f) That leaves 5 and 7. The algorithm continues by announcing that 5 is prime and removing it from the list.

g) Finally, 7 is announced and removed, and we're done.

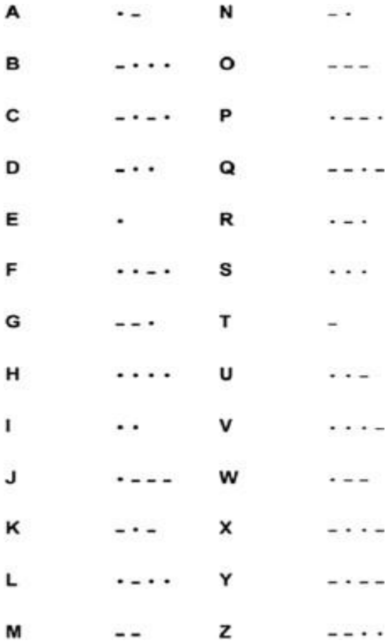
Write a program that prompts a user for n and then uses the sieve algorithm to find all the primesless than or equal to n. (Hint: Use list. Remove () method)

5. Write a function that returns the index of the smallest element in a list of integers. If the number of such elements is greater than 1, return the smallest index. Use the following header: def index Of Smallest Element (lst): Write a program that prompts the user to enter a list of numbers, invokes this function to return the index of the smallest element, and displays the index.

6. **(Count occurrences of numbers)** Write a program that reads an unspecified number of integers and finds the ones that have the most occurrences. For example, if you enter 2 3 40 3 5 4 –3 3 3 2 0, the number 3 occurs most often. Enter all numbers in one line. If not one but several numbers have the most occurrences, all of them should be reported. For example, since 9 and 3 appear twice in the list 9 30 3 9 3 2 4, both occurrences should be reported.

7. **Morse Code Encryption/Decryption Program:** Develop and test a Python program that allows a user to type in a message and have it converted into Morse code, and also enter Morse code and have it converted back to the original message. The encoding of Morse
code is given below:

Format the original message (containing English words) so that there is one sentence per line.



A	• –	N	– •
B	– • • •	O	– – –
C	– • – •	P	• – – •
D	– • •	Q	– – • –
E	•	R	• – •
F	• • – •	S	• • •
G	– – •	T	–
H	• • • •	U	• • –
I	• •	V	• • • –
J	• – – –	W	• – –
K	– • –	X	– • • –
L	• – • •	Y	– • – –
M	– –	Z	– – • •

Format the Morse code fi le (containing dots and dashes) so that there is one letter per line, with a blank line following the last letter of each word, and two blank lines following the end of each sentence (except the last).

RECOMMENDED BOOKS

1. A Downey, J. Elkner, and C. Meyers, “How to think like a computer scientist: learning with python. Green Tea Press”, Wellesley, Massachusetts, 2002.
2. J. Campbell, P. Gries, J. Montojo, and G. Wilson, “Practical programming: an introduction to computer science using Python”, Pragmatic Bookshelf, Second Edition, 2013.
3. A. B. Downey, “Python for software design: how to think like a computer scientist”, Cambridge University Press, 2009.
4. Z. A. Shaw, “Learn Python the Hard Way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code”, Addison-Wesley, 2013.

RECOMMENDED WEBSITES

1. http://swayam.gov.in

INSTRUCTIONAL STRATEGY

This is hands on practice based subject and topics taught in the class should be practiced in the Lab regularly for development of required skills among the students. This subject contains five units of equal weightage.

5.4 COMPUTER NETWORKS

L	P
3	4

RATIONALE

Global connectivity can be achieved through computer networks. After completing the diploma, student should have basic understanding of networking and its models. This subject will help the student in network setup and troubleshooting. It will further give exposure to the students about wireless networks and cloud computing.

COURSE OUTCOMES

After undergoing the subject, the students will be able to:

CO1: Learn about the basic concepts of networking models.

CO2: Handle different IP address classes.

CO3: Connect various networking devices.

CO4: Troubleshoot networking related issues.

CO5: Describe wireless networks and cloud computing.

DETAILED CONTENTS

UNIT I

NETWORKS BASICS

- Concept of network
- Models of network computing
- Networking models
- Peer-to –peer Network
- Client-Server Network
- LAN, MAN and WAN
- Network Services
- Topologies
- Switching Techniques

NETWORKING MODELS

- OSI model: Definition, Layered Architecture

Functions of various layers

- TCP/IP Model: Definition, Functions of various layers
- Comparison between OSI and TCP/IP model

UNIT II

TCP/IP ADDRESSING

- Concept of physical and logical addressing
- IPV4 addresses – Address space, Notations
- Classful Addressing- Different IP address classes, Classes & Blocks, Net-id & Host-Id, Masks, Address depletion
- Classless Addressing – Address blocks, Masks
- Special IP Addresses
- Subnetting and Supernetting
- Loop back concept
- Network Address Translation
- IPV4 Header
- IPV6 Header
- Comparison between IPV4 and IPV6

UNIT III

NETWORK ARCHITECTURE

Ethernet specification and standardization: 10 Mbps (Traditional Ethernet), 10 Mbps(Fast Ethernet) and 1000 Mbps (Gigabit Ethernet)

NETWORK CONNECTIVITY

- Network connectivity Devices
- NICs
- Hubs, Switches, Routers, Repeaters, Modem, Gateway
- Configuration of Routers & Switches

UNIT IV

NETWORK ADMINISTRATION

- Network Security Principles, Cryptography, using secure protocols

- Trouble Shooting Tools: PING, IPCONFIG, IFCONFIG, NETSTAT, TRACEROOT, Wireshark, Nmap, TCPDUMP, ROUTEPRINT
- DHCP Server
- Workgroup/Domain Networking

UNIT V
INTRODUCTION TO WIRELESS NETWORKS

- Introduction to wireless LAN IEEE 802.11, WiMax ad Li-Fi
- Wireless Security
- Introduction to bluetooth - architecture, application
- Comparison between bluetooth and Wifi

CLOUD COMPUTING

- Definition of Cloud Computing and advantages of Cloud Computing.
- Cloud Computing service model- SaaS, PaaS, Iaas.
- Deployment model-Private Cloud, Public Cloud, Hybrid, Community cloud.

PRACTICAL EXERCISES

1. Recognize the physical topology and cabling (coaxial, OFC, UTP, STP) of a network.
2. Recognition and use of various types of connectors RJ-45, RJ-11,BNC and SCST
3. Making of cross cable and straight cable
4. Install and configure a network interface card in a workstation.
5. Identify the IP address of a workstation and the class of the address and configure the IP Address on a workstation
6. Managing user accounts in windows.
7. Sharing of Hardware resources in the network.
8. Use of Netstat and its options.
9. Connectivity troubleshooting using PING, IPCONFIG, IFCONFIG
10. Installation of Network Operating System (NOS)

11. Demonstration of Cloud Computing in Labs or using Online Videos.

RECOMMENDED BOOKS

1. Computer Networks by Tanenbaum, Prentice Hall of India, New Delhi.
2. Data Communications and Networking by Forouzan, (Edition 2nd and 4th), Tata McGraw Hill Education Pvt. Ltd, New Delhi.
3. Data and Computer Communication by William Stallings, Pearson Education, New Delhi.
4. Local Area Networks by Peter Hudson.
5. Network+ Lab manual,- BPB Publications -by Tami Evanson.
6. Networking Essentials – BPB Publications New Delhi
7. Cloud Computing by Raj Kumar.

RECOMMENDED WEBSITES

1. http://swayam.gov.in

INSTRUCTIONAL STRATEGY

This is hands on practice based subject and topics taught in the class should be practiced in the Lab regularly for development of required skills among the students. This subject contains five units of equal weightage.

5.5PROGRAMME ELECTIVE – I

5.5.1 CLOUND COMPUTING

L	P
2	2

RATIONALE

This course offers a good understanding of cloud computing concepts and challenges faced in implementation of cloud computing. It also offers the concept of Virtualization along with security issues faced in the field of cloud computing.

COURSE OUTCOMES

After undergoing the subject, the students will be able to:

CO1: Explain core concepts of cloud computing paradigm.
CO2: Describe various Service and Deployment Models.
CO3: Detail SLA management in Cloud Computing.
CO4: Learn various trimmings and accessories sourced in garment production.
CO5: Illustrate the fundamental concepts of cloud storage and apply the concept of virtualization.

DETAILED CONTENTS

UNIT I
INTRODUCTION

Evolution of Cloud Computing, Cloud Computing Overview, Characteristics, Applications, Benefits, Challenges. Major Cloud Vendors/Service Provider in the world.

UNIT II
SERVICE AND DEPLOYMENT MODELS

Cloud Computing Service Models: Infrastructure as a Service, Platform as a Service, Software as a Service, Cloud Computing Deployment Models: Private Cloud; Public Cloud, Community Cloud, Hybrid Cloud, Major Cloud Service providers.

UNIT III
SERVICE LEVEL AGREEMENT (SLA) MANAGEMENT

Overview of SLA, Types of SLA, SLA Life Cycle, SLA Management Process.

UNIT IV
VIRTUALIZATION CONCEPTS

Overview of Virtualization, Types of Virtualization, Benefits of Virtualization, Hypervisors and its types.

UNIT V
CLOUD STORAGE &SECURITY

Storage as a Service, Benefits and Challenges, Storage Area Networks (SANs), Infrastructure Security, Network Level Security, Data Security & Privacy Issues, Legal Issues in Cloud Computing.

PRACTICAL EXERCISES

1. Introduction to Cloud Vendors: Amazon, Microsoft, IBM.
2. Setting up Virtualization using Virtualbox/VMWare Hypervisor
3. Introduction to OwnCloud and its features.
4. Installation and configuration of OwnCloud software for SaaS
5. Installing Open Source Cloud simulation software Cloud Sim.

RECOMMENDED BOOKS

1. Rajkumar Buyya, James Broberg, Andrzej Goscinski, “ Cloud Computing: Principles and Paradigms, Wiley.
2. Barrie Sosinsky, “ Cloud Computing Bible”, Wiley, 2011.
3. Judith Hurwitz, Robin Bloor, Marcia Kaufman,Fern Halper, “ Cloud Computing for Dummies”, Wiley, 2010.

RECOMMENDED WEBSITES

1. http://nptel.ac.in

2. https://ekumbh.aicte-india.org
3. http://swayam.gov.in

INSTRUCTIONAL STRATEGY

This is hands on practice based subject and topics taught in the class should be practiced in the Lab regularly for development of required skills among the students. This subject contains five units of equal weightage.