



Government Polytechnic Mumbai

(Academically Autonomous Institute of Maharashtra Government)
49, Ali Yawar Jung Marg, Kherwadi, Bandra (E)
gpmumbai@gpmumbai.ac.in

Programme: Computer Engineering

Fourth Semester

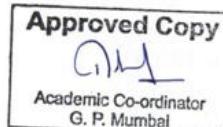
With effect from 2017-18

Course Code	Course Title	C/O	Teaching Scheme				Credits	Examination Scheme					
			L	P	TU	Total		Theory TH	TS	PR	OR	TW	Total
CO16309	Computer Networks	C	03	02	--	05	05	70	30	--	50*	--	150
CO16501	Software Engineering	C	03	--	--	03	03	70	30	--	--	--	100
CO16307	Java Programming	C	03	04	--	07	07	70#	30	50*	--	--	150
CO16401	Object Oriented Modeling & Designing	C	03	02	--	05	05	70	30	50	--	--	150
CO16205	Microprocessor	C	03	02	--	05	05	70	30	50	--	--	150
CO16402	Advance Web Page Design	C	02	04	--	06	06	--	--	50*	--	50	100
MG16502	Entrepreneurship and Development	C	01	--	02	03	03	--	--	25*	25	50	
Total			18	14	02	34	34	350	150	200	75	75	850

Abbreviations: C- Compulsory; O- Optional; L- Theory Lecture; P-Practical; TU-Tutorial; TH- Theory Paper; TS- Term Tests (02); PR-Practical Exam; OR-Oral Exam; TW- Term Work *Indicates assessment by Internal & External Examiners.

Indicates Online Examination

Academic Coordinator
(Dr. R. A. Park)



Head of Department

Principal

Programme : Diploma in Computer Engineering									
Course Code: CO16206				Course Title: Computer Networks					
Compulsory / Optional: Compulsory.									
Teaching Scheme and Credits				Examination Scheme					
TH	TU	PR	Total	TH	TS	PR	OR	TW	Total
03	-	02	05	70 (3 Hrs.)	30	--	50*	--	150

*Assessed by External Examiner.

Rationale:

A computer network or data network is a digital network which allows nodes to share resources. In computer networks, networked computing devices exchange data with each other. The networking part of this subject will stress on how to organize the network structure i.e. topology of the network as well as it will give the details about the hardware devices those are required to form a network. OSI (Open Systems Interconnection) model is introduced in the course, which is reference model for how applications can communicate over a network. Also it covers wired communication, protocol and wireless communication.

Course Outcomes:

Student should be able to:

CO1	Describe Basic Concept of Network., Classification of Network, transmission media
CO2	Identify the topology of the N/W
CO3	Select H/W devices used for networking
CO4	Configure wired and wireless LAN

Course Content Details:

Unit No	Topics / Sub-topics
1.	Introduction to data communication 1.1 Introduction to communication 1.1.1 Types of communication- i)Analog ii)Digital 1.1.2 Modulation : Definition 1.1.3 Modulation Types: Analog(AM,FM,PM), Digital(ASK,FSK,PSK) 1.2 Data Transmission: 1.2.1 Serial Transmission: Synchronous , Asynchronous 1.2.2 Parallel Transmission 1.3 Communication Modes: Simplex, Half Duplex, Full duplex

Computer Networks



CO16206

2. Network Concepts 2.1 Fundamentals of Computer Network- Definition, Need of Computer Network, Applications, Component of Computer Network: Host, Communication Subnet, NIC. 2.2 Computer Network Classifications- Classification of Network by their Geography.-PAN, CAN, LAN, MAN, WAN. 2.3 Classification of Network by their Component : Peer-to-Peer Network, client-Server-Based Network. 2.4 Network Topologies - Introduction, Types of Topology: i) Bus ii) Ring iii) Star iv) Mesh v) Tree vi) Hybrid.
3. Transmission Media and Networking Devices 3.1 Introduction – Need of Transmission Media, Selection Criteria. 3.2 Types of Transmission Media- <ul style="list-style-type: none"> 3.2.1 Guided Media: Cable Characteristics, Types of Cable-Twisted Pair Cable: Colour coding, Co-axial Cable, Fiber Optic Cable. 3.2.2 Network connectors: RJ45, BNC, Optical transceiver modules, connectors: SC, ST 3.2.3 Unguided media: Types of Communication Band: Microwave Communication, Radio wave Communication, Satellite Communication 3.3 Network Control / Connecting Devices : Role 3.4 Network Control devices: Network Interface Card , Hub, Repeater, Bridges, Switches, Router, Gateway, Modem. Firewall
4. OSI Reference Model 4.1 Introduction– Layered Architecture , Peer-to- Peer Processes- Interfaces between Layer. 4.2 Layers of the OSI Reference Model (Functions of each Layer & Protocols used) – Physical Layer, Data-Link Layer, Network Layer, Transport Layer, Session Layer, Presentation Layer, Application Layer.
5. Wired LANs: Point-to-Point Communication Protocol: 5.1 Fast Ethernet: MAC Sub layer, Physical Layer. Gigabit Ethernet: MAC Sub layer, Physical Layer. Ten-Gigabit Ethernet: MAC Sub layer, Physical Layer. 5.2 Ethernet IEEE Standards 802.3, Standard Ethernet: MAC Sub layer, Physical Layer, Changes in the Standard: Bridged Ethernet, Switched Ethernet, Full-Duplex Ethernet. 5.3 Multiple Access Random Access: ALOHA, CSMA, CSMA/CD, CSMA/CA 5.4 Framing, Transition Phases, Multiplexing, LCP, PAP, CHAP, NCP, IPCP, 5.5 Controlled access: Reservation, Polling, Token passing. Bit stuffing, byte stuffing, Channelization: FDMA, TDMA, CDMA.



6.	Wireless LANs : Wireless Communication Systems 6.1 Blue tooth: Architecture, Bluetooth Layers Connecting LANs, Backbone Networks and Virtual LANs 6.2 Wi-Fi Architecture, Wi-Fi Connecting LANs, WLL & LMDS 6.3 Introduction to Li-Fi
----	---

Suggested Specifications Table with Hours and Marks (Theory):

Unit No	Topic Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A Level	Total Marks
1	Introduction to data communication	05	06	04	--	10
2	Network Concepts	05	02	04	04	10
3	Transmission Media and Networking Devices	11	04	04	06	14
4	OSI Reference Model	11	06	08	--	14
5	Wired LANs: Point-to-Point Communication Protocol:	08	02	04	06	12
6	Wireless LANs : Wireless Communication Systems	05	--	04	06	10
Total		45	20	28	22	70

Legends: R- Remember; U-Understand; A- Apply and above levels (Bloom's revised Taxonomy).

Notes: This specification table shall be treated as a general guideline and actual distribution of marks may slightly vary from table. But the questions from each topic should be asked as per marks weightage. Numerical questions are to be asked only if specified.



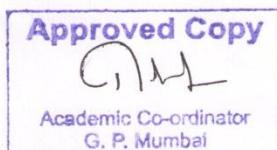
List of experiments/Assignments:

At least 12 experiments to be completed . Use Simulator (ex. Packet Tracer/ GNS3/ Network simulator/etc) wherever necessary.

Sr. No.	Unit	Experiment/Assignment	Approx. Hours
1.	2,3	Identify Components of Network in your Computer Network Lab.	2
2.	3	To Identify & observe Transmission Media.	2
3.	3	To Identify & observe Transmission Media connectors	2
4.	3	To Install Network Interface Card to locate MAC address of Computer.	2
5.	3	Prepare a network cable and Network Straight and Cross over Cable using RJ-45 connectors (to connect two hub/switch) and test by LAN Tester.	2
6.	2,3	Connect and configure different network control devices and Computers in Star/Ring Topology using Wired Media and any Network control Device. (Peer-to-Peer Network.)	2
7.	2,3	Connect and configure different network control devices and Computers in Mesh/Hybrid Topology using Wired Media and any Network control Device. (Peer-to-Peer Network.)	2
8.	3,5,6	Sketch wiring diagrams of network cabling considering existing computer lab of 20 systems	2
9.	2,3	Connect two or more different LANS using Bridge. (use simulator)	2
10.	5	Configure Point To Point Connection using ADSL Router (use simulator)	2
11.	5	Setup and configure LAN using fiber optics cable , connectors and suitable L2/L3 Switch	2
12.	5	To create scenario and observe the performance of network with CSMA/CA protocol and compare with CSMA/CD protocols using simulator	2
13.	6	Install & configure wireless LAN using Wi-Fi and configure Hotspot	2
14.	6	Install & configure wireless LAN using Bluetooth and configure Hotspot	2
15.	4	Use of basic networking commands	2
16.	3,4,5, 6	Identify and Trouble shoot the problem in any existing non functioning LAN	2
17.	All	Arrange Industrial Visit to observe resource sharing and deliver 5 minutes Seminar by each Student.	--
18.	All	Mini Project to be completed by the group of three or four students.	4
		Total	36

Note: If possible an industrial visit should be arranged or related videos should be shown.

Computer Networks



CO16206

References/ Books:

Sr. No.	Title	Author	Publisher
1	Data Communications and Networks	Achyut S. Godbole	Tata McGraw Hill
2	A.S.Tanenbaum	Computer Network	PHI
3	Data Communications and Networking (Forth Edition)	Behrouz A. Forouzan	Tata McGraw Hill
4	Computer Networking	Tularam M Bansod	Dreamtech,Wiley
5	Complete Reference Networking	Craig Zacker	Tata McGraw Hill

Web References:

<https://www.netacad.com/courses/packet-tracer-download/>
[http://library.aceondo.net/ebooks/Computer Science/Data Communication and Networking by Behrouz.A.Forouzan 4th.edition.pdf](http://library.aceondo.net/ebooks/Computer%20Science/Data%20Communication%20and%20Networking%20by%20Behrouz.A.Forouzan%204th.edition.pdf)

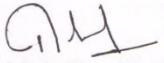
<http://web.cs.ucdavis.edu/~liu/289I/Material/book-goldsmit.pdf>

Course Curriculum Development Committee:**a. Internal Faculty**

- i. Smt. Varsha M Aswar
- ii. Ms. Pooja S. Chame
- iii. Mr. Nilesh Kitke

b. External Faculty

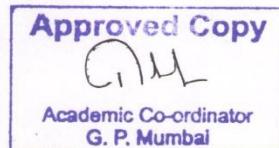
- i. Mr. Manish Salvi (Lecturer in Computer Engineering, Thakur Polytechnic, Kandivali, Mumbai)
- ii. Atul Mairale(HOD Computer Engg Dept., Kalavidyalay Poly)


 Academic Coordinator
 Govt. polytechnic Mumbai
 (Dr. R. A. Patil)


 Head of Department
 (Computer Engineering)


 Principal
 Govt. polytechnic Mumbai

Computer Networks



CO16206

Programme : Diploma in Computer Engineering									
Course Code: CO16501				Course Title: Software Engineering					
Compulsory / Optional: Compulsory.									
Teaching Scheme and Credits				Examination Scheme					
TH	TU	PR	Total	TH	TS	PR	OR	TW	Total
03	--	--	03	70(3 Hrs)	30	--	--	--	100

Rationale:

Software Engineering is an engineering discipline that is concerned with all aspects of software production. Further it is the systematic application of scientific and technological knowledge, methods and experience to the design, implementation, testing and documentation of software. This course intends to develop a systematic, disciplined approach to the development, operation and maintenance of software and help students to get acquainted with latest trends in Software Engineering.

Course Outcomes: Students will be able to

CO1	Explain Principles of Software Engineering
CO2	Apply Analysis Principles to S/W Project Development.
CO3	Apply Design Principles to S/W Project Development.
CO4	Write Project Management flow.
CO5	Describe basics of software Quality Assurance and Maintenance.



Course Contents:

Unit No.	Contents
1	Overview of Software Engineering <ul style="list-style-type: none"> 1.1 Definition of Software 1.2 Types, Characteristics and Applications of Software 1.3 Software Engineering- Definition, Need 1.4 Relationship between System Engineering and Software Engineering 1.5 Software Engineering- A Layered Approach 1.6 Software Development Generic Process Framework- Software Process, Software Product, Software Work-Product, Typical Umbrella Activities, Identifying A Task Set.
2	Prescriptive Process Models and Agile Methodology <ul style="list-style-type: none"> 2.1 Personal and Team Process Models (PSP and TSP) 2.2 Prescriptive Process Models: The Waterfall Model, V model, Incremental Process Model, Evolutionary Process Model: Prototyping. 2.3 Agile Software Methodology: <ul style="list-style-type: none"> 2.3.1 What is Agile Methodology 2.3.2 Difference between Prescriptive and Agile Process Model 2.3.3 Agility Principles 2.3.4 Agile Testing Methodology 2.3.5 Agile Process Model: Scrum 2.3.6 Scrum Process Flow 2.3.7 Introduction to Agile Tools: IceScrum 2.4 Introduction to DevOps
3	Software Engineering Practices <ul style="list-style-type: none"> 3. 1 Software Engineering Practices- Definition, Importance. 3. 2 Core Principles of Software Engineering 3. 3 Communication Practices <ul style="list-style-type: none"> 3.3.1 Concept, Need of Communication, 3.3.2 Statements and meaning of each Principle. 3.4 Planning Practices <ul style="list-style-type: none"> 3.4.1 Concept, Need of planning,



	<p>3.4.2 Statements and meaning of each Principle.</p> <p>3.5 Modeling Practices</p> <ul style="list-style-type: none"> 3.5.1 Concept of Software Modeling 3.5.2 Analysis Modeling Flow oriented Modeling(DFD, Data Dictionary, Decision Tables) Scenario based Modeling(What is Use Case, Purpose of Use Case, Use Case Diagram and its components) 3.5.3 Design Modeling Definition of Design Qualities of a Good Design Design Constraints. <p>3.6 Construction Practices</p> <ul style="list-style-type: none"> 3.6.1 Concept of Software Construction 3.6.2 Coding(Brief Introduction) 3.6.3 Validation(Brief Introduction) 3.6.4 Testing(Brief Introduction) <p>3.7 Software Deployment</p> <ul style="list-style-type: none"> 3.7.1 Concept of Delivery Life Cycle, Support Cycle and Feedback Cycle 3.7.2 Deployment Principles <p>3.8 SRS (Software Requirements Specification)</p> <ul style="list-style-type: none"> 3.8.1 Concept of SRS 3.8.2 General Format of SRS 3.8.3 Need of SRS 3.8.4 Case Study
4	<p>Software Project Management</p> <p>4. 1 The Management Spectrum –the 4 Ps and their Significance</p> <p>4. 2 Project Scheduling- Concept, Principles and Techniques(Gantt Chart, PERT, CPM,WBS)</p> <p>4. 3 Ways of Project Tracking</p> <p>4. 4 Risk Management</p> <ul style="list-style-type: none"> 4.4.1 Concept of Software Risks 4.4.2 Types of Software Risks <p>4. 5 Risk Assessment</p> <ul style="list-style-type: none"> 4.5.1 Risk Identification 4.5.2 Risk Analysis 4.5.3 Risk Prioritization <p>4.6 Risk Control</p> <ul style="list-style-type: none"> 4.6.1 RMMM Strategy



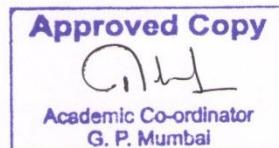
5	Software Testing and Quality Assurance 5. 1 Software Testing Fundamentals 5. 2 Testing Life-Cycle 5. 3 Testing Strategies 5.3.1 Verification and Validation 5.3.2 Defect Management Defect Life Cycle Bug Reporting 5. 4 Alpha and Beta Testing 5. 5 Introduction to White-Box and Black-Box Testing 5. 6 Software Quality Assurance- Definition and Activities for SQA 5. 7 Software Quality Factors.(McCall's) 5. 8 Quality Evaluation Standards
6	Software Maintenance 6. 1 Software Maintenance Definition 6. 2 Maintenance Characteristics and Activities 6.2.1 Corrective, Adaptive, Perfective, Preventive 6.3 Estimating Software Maintenance Cost 6.4 Maintenance Side Effects 6.5 Reverse Engineering and Re-Engineering



SUGGESTED SPECIFICATION TABLE WITH HOURS & MARKS (THEORY)

Unit No.	Unit Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A Level	Total Marks
1	Overview of Software Engineering	7	4	4	2	10
2	Prescriptive Process Models and Agile Methodology	8	4	6	2	12
3	Software Engineering Practices	10	6	4	6	16
4	Software Project Management	8	6	4	2	12
5	Software Testing and Quality Assurance	8	6	4	2	12
6	Software Maintenance	4	4	4	-	8
TOTAL		45	30	26	14	70

Software Engineering



CO16501

Reference Books:

Sr. No.	Book Title	Author	Publisher
1	Software Engineering- A Practitioner's Approach	Roger S. Pressman	Tata McGraw Hill
2	Software Engineering Concepts	Richard Fairly	McGraw Hill
3	Fundamentals of Software Engineering	Rajib Mall	Prentice Hall of India
4	A Concise Introduction to Software Engineering	Pankaj Jalote	Springer
5	Software Engineering	Jawadekar	Wiley India
6	Information Technology Project Management	Jack T. Marchewka	Wiley India

Web References:

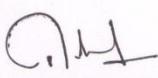
<http://www.win.tue.nl/~wstomv/edu/2ip30/references/>
https://www.tutorialspoint.com/software_engineering/

Course Curriculum Development Committee:**Internal Faculty:**

Ms. Pooja S. Chame (Lecturer in Computer Engineering, Government Polytechnic Mumbai)

External Faculty:

Mrs. Asawari Shiposkar (L and T Institute of Technology, Mahape ,Mumbai)



Academic Coordinator
(Dr. R. A. Patil)



Head of Department
(Computer Engineering)



Principal
Govt. Polytechnic Mumbai

Software Engineering



CO16501

Course Name:-Software Engineering

Course Code:- CO16501

CO Vs PO matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	3	2	-	-	1	1	1	1	2	2
CO2	3	3	1	1	2	1	1	2	3	2
CO3	3	3	1	1	2	1	1	2	3	2
CO4	2	2	3	2	3	2	1	2	2	1
CO5	2	2	1	1	2	-	1	1	2	1

CO Vs PSO matrix

	CO/POs	PSO1	PSO2	PSO3
CO1	Learn and Understand the Principles of Software Engineering	3	2	2
CO2	Apply Analysis Principles to S/W project Development	3	2	3
CO3	Apply Design Principles to S/W project Development	3	2	3
CO4	Understand Project Management Flow.	2	3	3
CO5	Learn basics of Software Quality Assurance and Maintenance	3	2	3

Unit Number and COs

Sr. No.	Unit No.	Topic Title	COs
1	1	Overview of Software Engineering	CO1: Learn and Understand the principles of Software Engineering
2	2	Prescriptive Process Models and Agile Methodology	CO1: Learn and Understand the principles of Software Engineering
3	3	Software Engineering Practices	CO2: Apply Analysis principles to S/W project Development. CO3: Apply Design principles to S/W project Development.
4	4	Software Project Management	CO4: Understand Project Management flow.
5	5	Software Testing and Quality Assurance	CO5: Learn basics of software Quality Assurance and Maintenance.
6	6	Software Maintenance	CO5: Learn basics of software Quality Assurance and Maintenance.

Software Engineering



CO16501

Programme : Diploma in Computer Engineering/Information Technology									
--	--	--	--	--	--	--	--	--	--

Course Code: CO16307				Course Title: Java Programming					
----------------------	--	--	--	--------------------------------	--	--	--	--	--

Compulsory / Optional: Compulsory.									
------------------------------------	--	--	--	--	--	--	--	--	--

Teaching Scheme and Credits				Examination Scheme					
-----------------------------	--	--	--	--------------------	--	--	--	--	--

TH	TU	PR	Total	TH	TS	PR	OR	TW	Total
03	--	04	07	70#	30	50*	--	--	150

Indicates Online Theory Exam.

* Assessed By External Examiner.

Rationale:

After having sufficient command on structured and object oriented programming, Computer Engineering students must learn the Java Programming language. Although Java language was developed two decades back, it occupies maximum share of technology in the market due to its continuous and adaptive evolution in the form of versions. Students should learn higher level programming using Java and make the use of latest features init, for better quality of software and betterment of the society in turn. In this course, emphasis is given on features of Java which have been and about to be stable for long time in the market such as Exception Handling, Collection Framework, Generics and Javadoc Creation, etc.

Course Outcomes:

Students will be able to

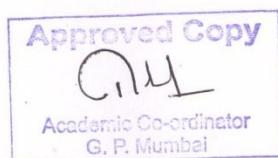
CO 1	Explain the Basic building blocks of a Java Program.
CO 2	Describe Object Oriented Features of Java Language.
CO 3	Explain the terms like Assignments, Garbage Collection and Operators in Java.
CO 4	Predict the flow of execution of Java programs using constructs, loops and exceptions.
CO 5	Develop a Java Application using I/O, Generics and Collection.
CO 6	Use Inner classes in Java Application Development.



Unit No.	Contents
1.	Declarations and Access Control 1.1 Identifiers &Keywords 1.2 Defining Classes 1.2.1 Source File Declaration Rules 1.2.2 Using the javac and java Commands 1.2.3 Using public static void main(String[] args) 1.2.4 Import Statements and the Java API 1.2.5 User Defined Packages 1.2.6 Static Import Statements 1.2.7 Class Declarations and Modifiers 1.3 Declaring Interfaces and interface constants 1.4 Declaring Class Members 1.4.1 Access Modifiers 1.4.2 Non-access Member Modifiers 1.4.3 Constructor Declarations 1.4.4 Variable Declarations 1.4.5 Declaring Enums 1.5 Java Version History 1.6 Creating First Java Application using IDE 1.7 Introduction to Javadocs 1.8 Creating javadocs for your own classes with the help of IDE
2.	Object Orientation 2.1 Encapsulation (Setters and Getters) 2.2 Inheritance and Polymorphism, Is-A, Has-A 2.3 Overriding / Overloading methods 2.4 Reference Variable Casting 2.5 Implementing an Interface 2.6 Legal Return Types 2.7 Constructors and Instantiation 2.8 Static Variables and Methods
3.	Assignments, Operators, Strings and Arrays 3.1 Use of Stack and Heap during Program Execution 3.2 Literals, Assignments, and Variables 3.3 Scope of Variables 3.4 Variable Initialization 3.5 Using Wrapper Classes and Boxing 3.6 Garbage Collection 3.7 Java Operators 3.8 Using String, StringBuilder, and StringBuffer classes and their methods



	3.9 Using Arrays 3.10 Arrays of Objects 3.11 Using ArrayList
4.	Flow Control, Exceptions, and Assertions 4.1 Loop Constructs 4.1.1 Enhanced for loop 4.1.2 Using break and continue 4.1.3 Unlabeled Statements 4.1.4 Labeled Statements 4.2 Handling Exceptions 4.2.1 Catching an Exception Using try and catch 4.2.2 Using finally 4.2.3 Propagating Uncaught Exceptions 4.2.4 Defining Exceptions 4.2.5 Exception Hierarchy 4.2.6 Handling an Entire Class Hierarchy of Exceptions 4.2.7 Exception Matching 4.2.8 Exception Declaration and the Public Interface 4.2.9 Rethrowing the Same Exception 4.3 Common Exceptions and Errors 4.3.1 Where Exceptions Come From 4.3.2 JVM Thrown Exceptions 4.3.3 Programmatically Thrown Exceptions 4.4 Multithreading 4.4.1 Thread Life Cycle 4.4.2 Extending Thread class 4.4.3 Implementing Runnable Interface 4.5 Overview of with the Assertion Mechanism
5.	I/O, Formatting, and Parsing 5.1 IO Streams and Console IO 5.2 File Navigation and I/O 5.2.1 Creating Files Using the File Class 5.2.2 Using FileWriter and FileReader 5.2.3 Combining I/O Classes 5.2.4 Working with Files and Directories 5.2.5 The java.io.Console Class 5.3 Files, Path, and Paths 5.3.1 Creating a Path 5.3.2 Creating Files and Directories 5.3.3 Copying, Moving, and Deleting Files 5.3.4 Retrieving Information about a Path 5.3.5 Normalizing a Path 5.3.6 Resolving a Path



	5.3.7 Relativizing a Path 5.4 Working with Dates, Numbers, and Currencies and Locale(Java 8 specific) 5.5 Parsing 5.6 Tokenizing 5.7 Formatting
6.	Overview of Generics and Collections 6.1 <code>toString()</code> , <code>hashCode()</code> and <code>equals()</code> methods 6.2 Collections 6.3 Using the Collections Framework 6.4 Generic Types
7.	Inner Classes and Other topics 7.1 Nested Classes and Inner Classes 7.2 Method-Local Inner Classes 7.3 Anonymous Inner Classes 7.4 A brief introduction to Lambda Expressions 7.5 Default Methods, Multiple Defaults, Static Methods on Interfaces 7.6 Classpath, Creating JAR Files and Executable JARs

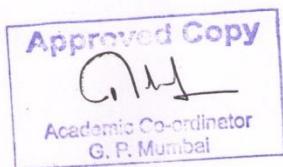
Suggested Specifications Table with Hours and Marks (Theory):

Unit No.	Title	Teaching Hours	Distribution of Theory Marks				Total Marks
			R Level	U Level	A Level		
1.	Declarations and Access Control	5	2	2	4		8
2.	Object Orientation	6	2	4	4		10
3.	Assignments, Operators, Strings and Arrays	6	2	4	4		10
4.	Flow Control, Exceptions, and Assertions	8	2	4	6		12
5.	I/O, Formatting, and Parsing	8	4	4	4		12
6.	Overview of Generics and Collections	7	2	4	4		10
7.	Inner Classes and Other topics	5	2	2	4		8
		Total	45	16	24	30	70



List of Experiments:

Sr. No.	Unit	Experiment/Assignment	Approx. Hours
1.	1	<p>Getting started with Java Application Development using IDE</p> <p>1.1 Check whether latest version of java (JDK 9 or at least JDK 1.8) is installed or not. If not then download and install it.</p> <p>1.2 Download and install the IntelliJ IDEA Community Edition/ NetBeans IDE 8.1/ Eclipse Neon or later version of IDE</p> <p>1.3 Create a Java Project/ Application in the IDE</p> <p>1.4 Create a Java class Person containing two variables name and yearOfBirth of appropriate data types, take inputs from the command line argument. Display the name and age of the person.</p> <p>1.5 Save the project and run it.</p> <p>1.6 Explore all the features (the menu) of the IDE. Learn about their use.</p>	2
2.	1	<p>2.1 Think-Pair-Share</p> <p>Think (15 mins) Enlist maximum known keyboard and mouse shortcuts in the IDE. e.g. insert code (inserting constructors, setter, getters, etc.), shortcuts for run, Refactor, breakpoint, etc.</p> <p>Pair (20 mins) Share your list of shortcuts to your partner and discuss what is being missed out. Update your list accordingly if required.</p> <p>Share (25 mins) Share your list with your batch mates and the faculty. Get it verified with the faculty. Prepare a consolidated list for entire batch and share it within the batch.</p> <p>2.2 Define the following classes/ interfaces with the help of above shortcuts(60 mins):</p> <ol style="list-style-type: none"> Person(id, name, dateOfBirth, age, street, city, pin : default and parameterized constructors and setters and getters) Department(id, name, dateOfEstablishment, headOfficeLocation, headId, numberOfWorkers : default and parameterized constructors and setters and getters) Point(x, y, z : default and parameterized constructors and setters and getters) Vehicle(registrationNumber, rcBookNumber, manufacturer, numberOfWorkers, vehicleType, model, numberOfWorkers : default and parameterized constructors and setters and getters) Laptop(imeiNumber, processorName, processorSpeed, primaryMemoryType, primaryMemoryCapacity, secondaryStorageType, secondaryStorageCapacity, screenResolution, screenType, isLED, listOfPorts, osInstalled : default and parameterized constructors and setters and getters) interface Taxable(public int cost(), public int percentGST()) 	2



3.	1	<p>Team-Pair-Solo with Javadocs (To be completed in a single lab session) Given: Open education resources like videos, etc. for Javadocs Creation by the faculty one week in advance.</p> <p>Team (35 mins)</p> <ol style="list-style-type: none"> 1. Make groups of 4 students from the batch. 2. Write down the definitions for any two of the five classes given in 2.2 in IDE with documentation comments: 3. Generate the javadoc for any two classes. 4. Share your work with other teams and get their feedback. 5. Modify the javadoc(s) accordingly if required. <p>Pair (35 mins)</p> <ol style="list-style-type: none"> 1. Repeat the activity in pairs for any two of the remaining classes. 2. Share your work with other pair in your team and get their feedback. 3. Modify the javadoc(s) accordingly if required. <p>Solo (30 mins)</p> <ol style="list-style-type: none"> 1. Repeat the same activity individually. 2. Share your work with other peer in your pair and get his/her feedback. 3. Modify the javadoc(s) accordingly if required. <p>Consolidate a final javadocs specification for the classes which is agreed unanimously by the students and the faculty. (20 mins)</p>	2
4.	2	<p>4.1 Check whether feature of Encapsulation has been followed in the classes in 2.2. If not make necessary changes in the classes.</p> <p>4.2 Define classes Car, Train and Truck with necessary attributes, constructors and methods. Implement IS-A relationship with the Class Vehicle.</p> <p>4.3 Define a class Gadget with necessary attributes, constructors and methods. Modify the class Laptop to extend the class Gadget.</p> <p>4.4 Create a subclass Employee of the class Person. HOD(id, name, street, city, pin, designation, salary, Department: necessary and/or overridden methods) to implement HAS-A relationship.</p> <p>4.5 Insert overloaded methods 'offer' in the class to implement the relationship 'Department offers Course' where Course is a class Course(courseCode, title, courseOutcomes, content)</p> <p>4.6 In main method, declare a reference variable vehicle of class Vehicle and create an object of class Car which will be referenced by vehicle. Call getName() method (in 4.2) on the object. (Hint: Reference Variable Casting)</p> <p>4.7 Modify the classes Vehicle and Gadget implement the interface Taxable. Hence override respective methods.</p> <p>4.8 Identify default and parameterized constructors in class Department. In main method initialize objects of class Department using all the overloaded constructors</p> <p>4.9 Modify the classes Car and Laptop to override the implemented methods in 4.7</p> <p>4.10 Modify the class Gadget to add a data member</p>	12



		gadgetCount such that its value will incremented as soon as a new object is initialized. Print its value after initializing 3 objects.	
5.	3	<p>5.1 Given 50 literals by faculty check whether each literal is legal or not. Justify your answer. Determine the type of the literal. Define all the legal literals in main method and display them.</p> <p>5.2 Given some sample code like</p> <pre>class Layout { // class static int s = 343; // static variable int x; // instance variable { x = 7; int x2 = 5; } // initialization block Layout() { x += 8; int x3 = 6;} // constructor void doStuff() { // method int y = 0; // local variable for(int z = 0; z < 4; z++) { // 'for' code block y += z + x; } } }</pre> <p>Predict the output and justify your answer. Type and modify the code to print values of each variable just before every closing curly brace. Try compile the class.</p> <p>5.3 Foo Corporation needs a program to calculate how much to pay their hourly employees. The Ministry of Labor requires that employees get paid time and a half for any hours over 40 that they work in a single week. For example, if an employee works 45 hours, they get 5 hours of overtime, at 1.5 times their base pay. The Government of Maharashtra requires that hourly employees be paid at least Rs. 250.00 an hour. Foo Corp requires that an employee does not work more than 60 hours in a week. Define a method calculateWedge() with necessary parameters and appropriate return type in the class Foo.</p> <p>5.4 Write a Java method to add 2D matrices.</p>	8
6.	4	<p>6.1 Create 2 arrays of objects of class Point (refer 2.2) and initialize them using basic and enhanced for loop respectively.</p> <p>6.2 Write a program to display all the digits of a n digit decimal number where n is input by user in command line.</p> <p>6.3 Write a program to demonstrate try, catch and finally blocks.</p> <p>6.4 Create a csv file which will contain 10 integers separated by comma in a spreadsheet. Read the file using class java.util.Scanner and display the sum of the numbers in the file. Handle all possible exceptions.</p> <p>6.5 Define your own exception and handle it.</p> <p>6.6 Create classes ThreadExtends and ThreadImplements respectively extending class Thread and interface Runnable. Print the name of the class while implementing run() method. Create objects of both the classes in main and invoke life cycle methods on the objects.</p>	10
7.	5	7.1 Write a Java program to take character, integer, and String input from user and display it.	12



		<p>7.2 Write a Java program to create, read and modify a file.</p> <p>7.3 Use the method you defined in 5.3 to store monthly wedges disbursed to each employee in a file. Every record should be stored in a new line.</p> <p>7.4 Use the output of 7.3 to create a file record of yearly wedges for a given financial year of Foo Corp.</p> <p>7.5 Create two objects of class Path viz., source and target. Perform the following operations</p> <ul style="list-style-type: none"> a. Create a file at source b. Copy a file from source to target c. Move a file from source to target d. Delete a file from source e. Retrieve information about source and target. f. Normalize the paths like "/a./b./c", ".classpath", "/a/b/c/..", "../a/b/c", "/Build_Project/scripts/.../My_Project/source" g. Resolve a Path of a file h. Relativize a path with respect to other path. <p>7.6 Write a program to display Dates, Numbers and Currencies in different formats.</p>	
8.	6	<p>8.1 Write a Java program to override <code>toString()</code> and <code>equals()</code> methods on a class.</p> <p>8.2 Write a program to use <code>ArrayList</code>.</p> <p>8.3 Given a <code>List<List<String>></code> write a program to convert it into a <code>List<String></code>.</p> <p>8.4 Write a generic method to exchange the positions of two different elements in an array.</p> <p>8.5 Write a generic method to count the number of elements in a collection that have a specific property (for example, odd integers, prime numbers, palindromes)</p>	8
9.	7	<p>9.1 Write a class Outer which will contain a nested class named <code>NestedClass</code>.</p> <p>9.2 Modify the Outer class to include an inner class named Inner.</p> <p>9.3 Write a program to create an anonymous class while overriding a method in an interface.</p> <p>9.4 Create JAR file in Terminal/ Command Prompt and in IDE</p> <p>9.5 Deploy a Java Project to create an Executable JAR</p>	4
10.	All	<p>5 online quizzes of at least 10 objective questions based on above contents shall be conducted with the help of any free open source learning management system.</p> <p>Each quiz will be of duration 20 minutes duration and must contain the questions of the following category:</p> <ol style="list-style-type: none"> 1. General Concept (MCQs) (4 questions) 2. What's wrong with the given code? (2 questions) 3. Guess the output of the given code (3 questions) 4. Write the code/ Fill in the missing code(1 question) 	
11.	All	<p>Assignment</p> <p>Solve total 200 Objective Questions based on the above contents.</p> <p>Note: Questions should be categorized as said in 10.</p>	
Total			60



Reference Books:

Sr. No.	Book Title	Author	Publication
1	Oracle Press OCA/OCP Java® SE 7 Programmer I & II Study Guide (Exams 1Z0-803 & 1Z0-804)	Kathy Sierra Bert Bates	McGraw Hill Education
2	Java™ The Complete Reference Ninth Edition	Herbert Schildt	McGraw Hill Education
3	Head First Java	Kathy Sierra Bert Bates	O'Reilly

Web References:

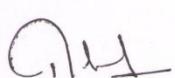
1. <https://docs.oracle.com/javase/tutorial/>
2. <http://www.angelikalanger.com/GenericsFAQ/JavaGenericsFAQ.html>
3. <https://www.youtube.com/playlist?list=PL9DF6E4B45C36D411>
4. http://spoken-tutorial.org/tutorial-search/?search_foss=Java&search_language=English

Course Curriculum Development Committee:**Internal Faculty**

Ms. Jijnasa S. Patil (Lecturer in Computer Engineering, Government Polytechnic Mumbai)

External Faculty

1. Mr. Suraj S. Bhosale (Lecturer in IT, Government Polytechnic, Jalgaon)
2. Mr. Vaibhav Vasani, (Assistant Professor, Shah & Anchor Kutchhi Engineering College, Mumbai)



Academic Coordinator
(Dr. R. A. Patil)



Head of Department
(Computer Engineering)



Principal
Govt. Polytechnic Mumbai



CO VsPO Matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	3	3	3	1	1	2	1	2	1	3
CO2	1	3	3	2	3	2	2	2	1	3
CO3	1	3	3	3	2	1	1	1	1	3
CO4	1	3	3	3	2	1	1	1	1	3
CO5	1	3	3	3	2	1	1	1	1	3
CO6	1	3	2	3	2	1	1	1	1	3

CO vs. PSO Matrix

COs		PSO1	PSO2	PSO3
CO1	Know the Basic building blocks of a Java Program.	3	2	3
CO2	Describe Object Oriented Features of Java Language.	3	3	3
CO3	Explain the terms like Assignments, Garbage Collection and Operators in Java.	3	3	3
CO4	Predict the flow of execution of Java programs using constructs, loops and exceptions.	3	3	3
CO5	Develop a Java Application using I/O, Generics and Collection.	3	3	3
CO6	Use Inner classes in Java Application Development.	3	3	3

Unit Number and COs

Sr. No.	Unit No.	Topic Title	COs
1	1	Declarations and Access Control	CO 1
2	2	Object Orientation	CO 2
3	3	Assignments, Operators, Strings and Arrays	CO 3
4	4	Flow Control, Exceptions, and Assertions	CO 4
5	5	I/O, Formatting, and Parsing	CO 5
6	6	Overview of Generics and Collections	CO 5
7	7	Inner Classes and Other topics	CO 6



Programme : Diploma in Computer Engineering									
Course Code: CO16401			Course Title: Object Oriented Modeling and Design						
Compulsory / Optional: Compulsory.									
Teaching Scheme and Credits				Examination Scheme					
TH	TU	PR	Total	TH	TS	PR	OR	TW	Total
03	-	02	05	70(3Hrs)	30	50	-	-	150

Rationale:

Modelling plays an important role in designing the application. Graphical notations used to model a variety of systems help the designer to visualize a problem before implementation. Object Oriented Modelling is more than just a way of programming. The real world scenarios and customer needs are mapped to models for an implementation. The mapping builds the bridge between user and model for improvising the performance of system. It facilitates to build a language independent design. Modelling is an essential part of large software projects. It enable user to implement the structure virtually for verifying user requirement are meet. So that final structure will be accurate.

Course Outcomes:

Student should be able to:

CO1	Ellaborate object oriented concepts
CO2	Implement high level requirement into modeling concepts
CO3	Map user's requirement with model.
CO4	Create custom UML profile to accurately model different system domains

Course Contents:

Unit No	Contents
1	Introduction to Object Oriented Modeling 1.1 Object Orientation 1.2 Object Oriented Development - OO methodology, Three Models 1.3 OO Themes: Abstraction, Encapsulation, Combining data and behavior 1.4 Modeling as Design techniques - Brief overview of OMT by Rumbaugh, Importance of Modeling, Four principles of Modeling 1.5 Introducing to the UML – overview, conceptual model, architecture, software development lifecycle
2	Object and Class Concepts: 2.1 Objects, Classes 2.2 Class Diagrams 2.3 Values and Attributes



	2.4 Operations and Methods 2.5 Link and Association concepts -Links and Associations, Multiplicity, Association and Names, Ordering ,Association Classes, Qualified Association 2.6 Generalization and Inheritance -Use of Generalization, Sample Class Model
3	Multiplicity , Aggregation and Object Modeling 3.1 Multiplicity, Aggregation 3.2 Aggregation Versus Association 3.3 Propagation of operations 3.4 Multiple Inheritance ,Metadata and Constraints-Metadata, Constraints on objects and links 3.5 Object modeling 3.6 Object instances 3.7 Sample Object Model
4	Basic Behavioral Modeling 4.1 Use case Diagram 4.1.1 Notations for Use case diagram – use cases, Actors, Communication lines, System boundaries 4.1.2 Use case relationships - Include and extend, Use case generalization 4.1.3 Sample use case diagrams 4.2 Sequence diagram 4.2.1 Notations for Sequence diagram – Objects / Participants, Time, events, Activation Bars , signals , message arrows, synchronous and asynchronous messages, return message, create and destroy message 4.2.2 Structured control - optional, conditional, parallel, loop execution 4.2.3 Sample sequence diagrams 4.3 Collaboration Diagram 4.3.1 Terms and Concepts, Modeling techniques.
5	Advanced Behavioral Modeling 5.1 Activity Diagram 5.1.1 Notations for Activity Diagram - Actions and Activity nodes, initialization and completion, Decisions, Join and fork 5.1.2 Doing multiple tasks at the same time -Swim lanes 5.1.3 Sample Activity Diagram 5.2. State Diagram 5.2.1 Notations for State diagram - initial state, final state, transitions and conditions, activity, event, 5.2.2 Nested state diagram, concurrent / composite state diagram



	5.2.3 Sample state diagram
6	<p>Architecture Modeling</p> <p>6.1 Component Diagram</p> <p>6.1.1 Notations for component Diagram - component and interfaces, ports, connectors</p> <p>6.1.2 Sample Component Diagram</p> <p>6.2 Deployment Diagram</p> <p>6.2.1 Notations for Deployment diagram - nodes, artifacts, node instances, communication between nodes</p> <p>6.2.2 Sample Deployment diagram</p>

Suggested Specification Table With Hours & Marks (Theory):

Unit No.	Unit Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A Level	Total Marks
1	Introduction to Object Oriented Modeling	08	04	06	02	12
2	Object and Class Concepts	08	04	06	02	12
3	Multiplicity, Aggregation and Object Modeling	10	06	06	04	16
4	Basic Behavioral Modeling and Sequence diagram	08	06	04	02	12
5	Advanced Behavioral Modeling and State Diagram	05	02	02	04	8
6	Architecture Modeling and Deployment Diagram	06	02	04	04	10
Total		45	24	28	18	70

Legends: R- Remember; U-Understand; A- Apply and above levels (Bloom's revised Taxonomy).

Notes: This specification table shall be treated as a general guideline and actual distribution of marks may slightly vary from table. But the questions from each topic should be asked as per marks weightage. Numerical questions are to be asked only if specified.

List of experiments/Assignments:

At least 12 programs to be completed. Use software (ex. SmartDraw / StarUML community edition etc) wherever necessary.

Sr. No	Unit	Experiments/Assignments	Hours
1.	1	Install SmartDraw OR StarUML OR alike open source software.	02
2.	1	Operate SmartDraw with its functionality to draw various UML notations.	02
3.	2	Draw class diagram with notations for following systems. a. Library management system b. Hospital management system	02
4.	2	Draw class diagram with generalization and inheritance for following systems. a. Online railway reservation system b. Hotel management system	02
5.	3	Draw object diagram with notations for following systems. a. Online railway reservation system b. Hotel management system	02
6.	3	Draw object diagram with multiple inheritance for following systems. a. Library management system b. Hospital management system	02
7.	4	Draw use case diagram with notations for following systems. a. Book bank facility in College library	02
8.	4	Draw collaboration diagram with notations for following systems. a. Online Flight Reservation system b. Exam registration System	02
9.	4	Draw sequence diagram with notations for following systems. a. Online payment portal b. Recruitment System	02
10.	5	Draw activity diagram with notations for following systems a. Online Course registration b. E-Commerce Application	02
11.	5	Draw state diagram with notations for following systems. a. Bank ATM b. Online web portal for shopping Electronics gadgets	02



12.	6	Draw component diagram with notations for following systems a. Library management system b. Hospital management system	02
12.	6	Draw deployment diagram with notations for following systems a. Online bus/railway reservation system b. Hotel management system	02
14	1 to 6	Develop mini project for any one of the following: (Draw all UML diagrams) 1. Passport Automation system 2. Exam Registration 3. Stock maintenance system 4. Online course reservation system 5. Foreign Trading systems 6. Conference Management System 7. BPO Management system 8. Credit card processing 9. E-book management system 10. Software personnel management System	06
TOTAL			30

References Books:

Sr. No.	Book Title	Author	Publisher
01	Object oriented modeling and design with UML 2.0 (second edition)	Blaha and Rumbaugh	Pearson
02	System Analysis and design - an Object oriented approach with UML	Dennis, Wixom, Tegarden	5th Edition, Wiley publication
03	Software Testing Principle, Techniques and Tools	M G Limaye	TMH
04	The Unified Modeling Language User Guide	Grady Booch, James Rumbaugh, Ivor Jacobson	Second Edition, Addison Wesley Object Technology Series.
05	UML 2 and the Unified Process: Practical Object Oriented Analysis and design	Jim Arlow	Second Edition, , Addison Wesley Object Technology Series
06	Design Patterns: Reusable elements of object oriented software	Erich Gamma and others	Pearson Education Series. Software Engineering

Object Oriented Modeling and Design



CO16401

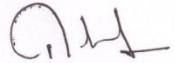
Course Curriculum Development Committee:

a. **Internal Faculty:**

- 1 . Mrs Anuradha Bhatt Oza,(Lecturer, Government Polytechnic,Mumbai)
2. Ms Ruchira Varekar(Visiting Faculty,Computer Engineering, Govt.Polytechnic Mumbai)

b. **External Faculty:**

- Mrs. Shipa Kabra.(Lecturer ,Government College of Engineering, Aurangabad)



Academic Coordinator
Govt. polytechnic Mumbai
(Dr. R. A. Farid)



Head of Department
(Computer Engineering)



Principal
Govt. polytechnic Mumbai



Course Name:-Object Oriented Modeling and Design

Course Code:- CO16401

CO Vs PO matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	3	2	-	-	1	1	1	1	2	2
CO2	3	3	1	1	2	1	1	2	3	2
CO3	3	3	1	1	2	1	1	2	3	2
CO4	2	2	3	2	3	2	1	2	2	1
CO5	2	2	1	1	2	-	1	1	2	1

CO Vs PSO matrix

CO	PO	CO/POs			PSO1	PSO2	PSO3
		CO1	CO2	CO3			
CO1	Learn and Understand the principles of Software Engineering				3	2	2
CO2	Apply Analysis principles to S/W project Development.				3	2	3
CO3	Apply Design principles to S/W project Development.				3	2	3
CO4	Understand Project Management Flow.				2	3	3
CO5	Learn basics of Software Quality Assurance and Maintenance.				3	2	3

Unit Number and COs

Sr. No.	Unit No.	Topic Title	COs
1	1	Introduction to Object Oriented Modeling	CO1: Learn and Understand the principles of Software Engineering
2	2	Object and Class Concepts	CO1: Learn and Understand the principles of Software Engineering
3	3	Multiplicity , Aggregation and Object Modeling	CO2:Apply Analysis principles to S/W project Development. CO3:Apply Design principles to S/W project Development.

Object Oriented Modeling and Design

Approved CopyAcademic Co-ordinator
G. P. Mumbai

CO16401

4	4	Basic Behavioral Modeling	CO4:Understand Project Management flow.
5	5	Advanced Behavioral Modeling	CO5:Learn basics of software Quality Assurance and Maintenance.
6	6	Architecture Modeling	CO5:Learn basics of software Quality Assurance and Maintenance.



Programme : Diploma in CO / IT									
Course Code: CO16205				Course Title: Microprocessor and Microcontroller					
Compulsory / Optional: Compulsory									
Teaching Scheme and Credits				Examination Scheme					
TH	TU	PR	Total	TH	TS	PR	OR	TW	Total
3	-	2	5	70 (3 Hrs.)	30	50			150

Rationale:

Microcontroller is heart of all domestic, industrial, consumer goods and other high end products. Automation in every field of life is being used and microcontroller is inbuilt element of these systems. The purpose of this subject is to cover the underlying concepts and techniques used in Microprocessor and microcontroller. It also covers interfacing of microcontroller with various input and output devices such as ADC, DAC, display and stepper motor. Students will learn assembly language programming of 8085 and 8051. They will learn how to design microcontroller based systems for various applications.

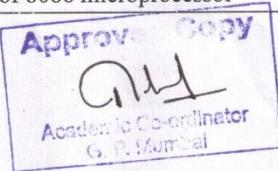
Course Outcomes:

After the completion of the course student will be able to

CO1	Describe architecture and pin configuration of 8086 microprocessor.
CO2	Develop assembly language programs of 8086
CO3	Describe architecture and pin configuration of 8051 microcontroller.
CO4	Develop assembly language programs of 8051
CO5	Design and develop microprocessor and microcontroller based system
CO6	Describe architecture of i7 processor

Course Content Details:

Unit No	Topics / Subtopics
1	MICROPROCESSOR: 1.1 Introduction to single board microcomputer Block diagram of microcomputer Elements of microcomputer Different type of Buses: Address, data and control bus 1.2 Evolution of microprocessor 1.3 Architecture of 8086 microprocessor



	1.4 Pin configuration of 8086 microprocessor
2	INSTRUCTION SET AND PROGRAMMING OF 8086: 2.1 Addressing modes of 8086 2.2 Instruction set of 8086 2.3 Assembly language programming of 8086 (Only Addition, Subtraction, Arranging numbers in ascending and descending orders) 2.4 Instruction cycle, machine cycle, timing diagram of memory read, memory write operation
3	8051 MICROCONTROLLER: 3.1 Distinguish microcomputer ,microprocessor and microcontroller 3.2 Architecture of 8051 3.3 Memory organization of 8051 3.4 Pin configuration of 8051 and features
4	INSTRUCTION SET AND PROGRAMMING OF 8051: 4.1 Addressing mode 4.2 Instruction set 8051 -Data Transfer Instruction set -Arithmetic Instruction set -Logical Instruction set -Branching Instruction set -Boolean Instruction set -Assembler directive-ORG,DB,EQU,END,CODE,DATA 4.3 Assembly language programming of 8051
5	PARALLEL PORTS AND SERIAL COMMUNICATION : 5.1 Basics of I/O concept and I/O port operation 5.2 Basics of serial data communication: SCON, SBUF 5.3 Modes of serial communication 5.4 RS232 serial communication
6	INTERRUPT AND TIMER : 6.1 Basics of interrupt and timer 6.2 Interrupt structure, polling ,SFR-IE,IP 6.3 Timer and counter programming 8051 6.4 Timer /counter logic and modes
7	MEMORY AND I/O INTERFACING of 8051: 7.1 Memory interfacing -interfacing external RAM and ROM 7.2 I/O Interfacing Interfacing of keyboard, LED, 7 segment display, stepper motor 7.3 Design microcontroller based system (Level controller, traffic controller)
8	ADVANCED PROCESSORS 8.1 Introduction to i7 architecture



Suggested Specifications Table with Hours and Marks (Theory):

Unit No	Topic Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A Level	Total Marks
1	MICROPROCESSOR	06	04	04		08
2	INSTRUCTION SET AND PROGRAMMING OF 8086	06	02	02	06	10
3	8051 MICROCONTROLLER	06	04	04		08
4	INSTRUCTION SET AND PROGRAMMING OF 8051:	09	02	02	10	14
5	PARELLEL PORTS AND SERIAL COMMUNICATION :	04	02	04	02	08
6	INTERRUPT AND TIMER :	06	02	02	04	08
7	MEMORY AND I/O INTERFACING:	06		02	08	10
8	ADVANCED PROCESSOR	02		04		04
		Total	45	16	24	30
						70

Legends: R- Remember; U-Understand; A- Apply and above levels (Bloom's revised Taxonomy).

Notes: This specification table shall be treated as a general guideline and actual distribution of marks may slightly vary from table. But the questions from each topic should be asked as per marks weightage. Numerical questions are to be asked only if specified.

List of Practicals :- Any Ten

1	Understand 8086 and 8051 development board and simulation software
2	8086 Assembly language programming for Addition and subtraction of two 8 bit numbers
3	8086 Assembly language programming for arranging numbers in ascending and descending order.
4	For 8051 Microcontroller Develop and simulate assembly language program for arithmetic operations as addition, subtraction, multiplication, division.
5	For 8051 Microcontroller Develop and simulate assembly language program for Block transfer and Block Exchange with external memory.
6	For 8051 Microcontroller Develop and simulate assembly language program for finding smallest/largest numbers and arranging the numbers in ascending/descending order.
7	Develop and simulate assembly language program to generate square and rectangular wave on port pin of 8051 using timer.
8	Develop and simulate assembly language program to ON/OFF LED using a key connected at ports of 8051.
9	Interface seven segment display to 8051 and develop, simulate an assembly language program to design UP/DOWN counter (using Timer Interrupts)
10	Interface stepper motor to 8051 and develop program to rotate motor in clockwise direction.
11	Interface 8 bit DAC to 8051. Develop and download an assembly language program

Microprocessor and Microcontroller



CO16205

	to generate at least two different waveforms using DAC and convert digital data into analog using DAC.
12	Interface 8 bit ADC to 8051. Develop and download an assembly language program to generate at least two different waveforms using ADC and convert digital data into analog using ADC.
13	Develop and simulate assembly language program for Level controller.
14	Develop and simulate assembly language program for Traffic controller.

Reference Books:

Sr. No.	Book Title	Author	Publication
1	8051 Microcontroller architecture programming & application.	K. J. Ayala	EEE/ Prentice Hall of India
2	The 8051 microcontroller & embedded system	Mohmad-ali-mazidi, Janice-Gelispe-mazidi , Roline D. Mckinlay	Pearson / Prentice hall
3	Microcontroller principal & application	Ajit Pal	Prentice Hall of India
4	Microcontroller theory & application.	Ajay Deshmukh	Tata McGraw- Hill
5	Microcontroller Architecture, programming, interfacing, & system design	Rajkamal	Pearson
6	Microprocessor & Interfacing (Programming & Hardware)	Douglas Hall	The McGraw-Hill Companies
7	Advanced Microprocessor & Pheripherals	A.K. Ray & K.M. Bhurchandi	The McGraw-Hill Companies

Course Curriculum Development Committee:**a. Internal Faculty**

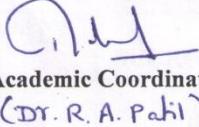
Dr. R. A. Patil (Sel. Grade Lecturer, Electronics Engineering, Govt. Polytechnic Mumbai)

Ms. S. R. Nagargoje (Lecturer, Electronics Engineering, Govt. Polytechnic Mumbai)

Mr. N. R. Kitke(Visiting Faculty, Computer Engineering, Govt. Polytechnic Mumbai)

b. External Faculty

Dr. Uday Khot (Lecturer, Electronics Engineering, St. Francis Institute of technology, Mumbai)



Academic Coordinator
(Dr. R. A. Patil)



Head of Department
(Name of Discipline)



Principal
Govt. Polytechnic Mumbai



Programme : Diploma in Computer EngineeringCourse Code: **CO16402** Course Title: **Advanced Web Page Design****Compulsory / Optional: Compulsory.**

Teaching Scheme and Credits				Examination Scheme					
TH	TU	PR	Total	TH	TS	PR	OR	TW	Total
02	-	04	06	--	--	50*	--	50	100

* Assessed By External Examiner

Rationale :

This course encompasses many different skills and disciplines in the production and maintenance of websites. Technology is based on dot net frame work, which supports many languages so that application designed in one language can be connected/interfaced with this frame work hence it is more flexible and advanced. This course helps students to enhance their skills & competencies in web design software.

Course Outcomes :

The student will be able to:

CO1	Describe the basics of .Net Framework
CO2	Use basic and advance .Net controls.
CO3	Interface back-end and front-end.
CO4	Build applications integrated with .NET Framework.
CO5	Demonstrate an understanding of the use of databases in website development.



Course Contents:

Unit No	Contents
1	<p>Introduction</p> <ul style="list-style-type: none"> 1.1 Introduction to Microsoft .Net Framework. 1.2 Building blocks in .Net 1.3 Drawback of previous languages. 1.4 .Net framework: components of .Net framework, Advantages, requirement of .Net
2	<p>Introduction and implementation of VB.Net</p> <p>2.1 Introduction to VB.Net</p> <ul style="list-style-type: none"> • VB.Net overview, • Difference between VB and VB.Net <p>2.2 Implementation of VB.Net</p> <ul style="list-style-type: none"> • Features.,VB.Net IDE. • Data Types, Loops, Control structures, Cases, Operators. • Creating forms. • Procedures and functions. • Form controls. • Error Provider • ComboBox • MonthCalendar • RadioButton • TextBox • CheckBox • CheckedListBox • DateTimePicker <p>2.3 Implementation of OOP</p> <ul style="list-style-type: none"> • Creation of class and objects. • Inheritance. • Constructors. • Exception handling.



3	<p>Introduction to ADO.Net and data manipulation</p> <p>3.1 Introduction to ADO.Net</p> <ul style="list-style-type: none"> • What is database? • Writing XML file. • ADO.Net architecture. • Creating connection. • Dataset and Data reader. • Types of Data adapter and ADO controls. • Reading data into dataset and data adapterBinding data to controls. • Data table and Data row. <p>3.2 Accessing and manipulating data</p> <ul style="list-style-type: none"> • Selecting data. • Insertion, deletion, updation, sorting. • How to fill dataset with multiple tables.
4	<p>Introduction to ASP.Net</p> <ul style="list-style-type: none"> • Difference between ASP and ASP.Net • Introduction to IIS. • ASP.Net IDE. • Creation of web forms using web form controls. • ASP.Net page life cycle. • Web Server Controls. • Validation Controls.
5	<p>ASP.Net objects and components</p> <p>5.1 ASP.Net Objects</p> <ul style="list-style-type: none"> • Response. • Server. • Application. • Session. • Request



	<ul style="list-style-type: none"> ASP.Net scope, state, view state, post back and configuration. <p>5.2 How to use objects?</p> <ul style="list-style-type: none"> Object creation: Scripting, Drive, folder, file. How to use Application object. - Events - Methods and collection. - Example. How to use session object : enabling and disabling of session, Event, properties, methods, collection. Example. <p>5.3 Server components :</p> <ul style="list-style-type: none"> Ad rotator, Content linker, Browser capabilities. Use and creation of global.asax file.
6	<p>ADO.Net and Data Manipulation</p> <p>6.1 ADO.Net in ASP.Net</p> <ul style="list-style-type: none"> Connection. Dataset and data reader. Data table and Data row. Web.config introduction. Binding data with data grid. Accessing and manipulating data. <p>6.2 ADO.Net :</p> <ul style="list-style-type: none"> Server control templates and Data binding techniques Understand data access in .Net using ADO.Net Understand various Server Control Templates available for Data Binding using Repeater Control Data List control, Data Grid Controls, FormView Control, DetailView Control



List of Practicals :-

Sr. No.	Unit	Experiments/Assignments	Approx. Hours
1	1	Introduction to .Net framework & environment setup.	02
2	2	a) Design Login form with validation. b) Design Registration form with validation of email address, date of birth, telephones and mobile numbers etc.	04
3	2	Design form, make it a class, create its object and access it from another form	04
4	2	Design student class, marks class, inherits it in result class and access it using form.	02
5	3	Design mark sheet of student using XML file and dataset.	02
6	4	Generation of database (data table) of employee or student with help of data tables of .Net	02
7	4	Design registration form of college using text box, text area, radio list, check list, button etc. using Autopostback property.	08
8	5	Simple application for following function: (1) Login (2) Surfing (3) Logout taking into considerations (Application, Session, Server object, global.asa file and their events, methods and collection), also demonstrates enabling and disabling of session.	08
9	5	Creation of file, entry, reading data from a file.	04
10	5	Using components create: (1) Advertisement (using Ad rotator) (2) Book example (using Next function) (3) find capabilities of browser (Browser object capabilities)	04
11	6	Design employee details with help of database (back-end) using data adapter, data reader and datasets. Use data grid to display result	04
12	4&6	Online application (student, employee, product, shopping mall) (a) Using dataset, data reader. (b) Same application using data table and data row. (use data grid to display data) (c) Bind the data to data grid using properties / templates. (d) Display details (student, employee, product, etc.) using data list. (4 cols per line)	08
13	All	Mini Project	08
Total			60

*Mini Project should be assigned by subject teacher.



Reference Books:

Sr. No.	Book Title	Author	Publication
01	Prog. In VB.Net	Anita & Bradely	Tata McGraw Hill
02	ASP.net	Dave Mercer	Tata McGraw Hill
03	Beginning VB.Net 2003	--	Wrox Publication
04	Designing Application with Microsoft VB.net	Robert LandLizer	Tata Mc Grow Hill
05	Beginning ASP.Net	--	Wrox Publication
06	Prog. In VB.net	Grun grundgier	O'Reilly
07	.Net Frame Work Essential	Thuan Thai and Hoang Q. Lam	O'Reilly

Reference Links:

1. www.vbtutor.net
2. www.tutorialspoint.com/vb.net
3. www.tutorialspoint.com/ASP.Net
4. www.asp.net

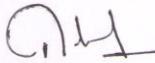
Course Curriculum Development Committee:+

1. Internal Faculty

Mrs. R. V. Molawade (Lecturer in Computer Engineering, Govt. Polytechnic, Mumbai)

2. External Faculty

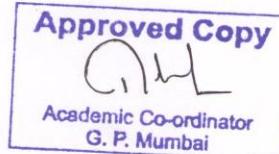
Mrs. Nisha Wartha (Lecturer in Information Technology, Govt. Polytechnic, Thane)


Academic Coordinator
Govt. Polytechnic Mumbai
(Dr. R. A. Paki)


Head of Department
(Computer Engineering)


Principal
Govt. Polytechnic Mumbai

Advanced Web Page Design



CO16402

Course: Advanced Web Page Design

Course Code:CO16402

CO Vs PO matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
1	3	3	2	3	2	2	2	1	1	3
2	3	3	3	3	2	2	2	3	3	3
3	3	3	2	3	3	2	2	2	2	3
4	3	3	3	3	3	2	2	3	3	3
5	2	2	2	2	3	2	2	2	2	3

CO Vs PSO matrix

CO	PSO1	PSO2	PSO3
Understand the basics of .Net Framework	2	1	1
Use basic and advance .Net controls.	3	2	2
Interface back-end and front-end.	3	2	3
Build applications integrated with .NET Framework.	3	3	3
Demonstrate an understanding of the use of databases in website development.	2	2	2

Approved Copy

Academic Co-ordinator
G. P. Mumbai

