ITM (SLS) Baroda University

School of Computer Science Engineering & Technolgy

Teaching Scheme of B.Tech (Cyber Security & Networking) ALL SEMESTERS EFFECTIVE FROM 2022-23

Semester 1

Sr#	Course Code	Course Name	Teaching Scheme Per Week (in Hrs.)			Hours	Credit
			L	**T**	**P**		
1	C2110C1	Programming in Python-1	3	0	4	7	5
2	S2110C1	Discrete Mathematics with Python	4	0	2	6	5
3	C2110C2	R Programming for Data	3	0	2	5	4
4	C2110C3	Web Technologies: HTML, CSS, JS, PHP	2	0	4	6	4
		Total	**12**	**0**	**12**	**24**	**18**
		Project-Based Learning					
5	C2110A1	PBL1.1:- MIT Inventor	0	2	0	2	2
6	C2110A2	PBL1.2:- Intel 8085	0	2	0	2	2
		Total	**12**	**4**	**12**	**28**	**22**

Semester 2

Sr#	Course Code	Course Name	Teaching Scheme Per Week (in Hrs.)			Hours	Credit
			L	**T**	**P**		
1	C2210C1	Programming in C	3	0	4	7	5
2	C2210C2	Data Structures and Algorithms-1	3	0	2	5	4
3	C2210C3	Computer Graphics	4	0	2	6	5

4	C2210C4	Programming in Python-2	2	0	4	6	4
		TOTAL	**12**	**0**	**12**	**24**	**18**
		Project-Based Learning					
5	C2230A1	PBL 2.1: Desktop OS Security: Windows and Ubuntu	0	2	0	2	2
6	C2230A2	PBL 2.2: Application Security	0	2	0	2	2
		Total	**12**	**4**	**12**	**28**	**22**
Semester 3							
Sr#	**Course Code**	**Course Name**	**Teaching Scheme Per Week (in Hrs.)**			**Hours**	**Credit**
			L	**T**	**P**		
1	C2310C1	Object-oriented Programming using Java	3	0	4	7	5
2	C2310C2	Computer Architecture	3	0	2	5	4
3	C2310C3	Systems Software	3	0	2	5	4
4	C2310C4	Database Management Systems	3	0	4	7	5
		TOTAL	**12**	**0**	**12**	**24**	**18**
		Project-Based Learning					
5	C2330A1	PBL-3.1: Linux Basics and Kali Pen Testing Tools	0	2	0	2	2
6	C2330A2	PBL-3.2: Basic Network Administration	0	2	0	2	2
		TOTAL	**12**	**4**	**12**	**28**	**22**
Semester 4							
Sr#	**Course Code**	**Course Name**	**Teaching Scheme Per Week (in Hrs.)**			**Hours**	**Credit**
			L	**T**	**P**		

			L	T	P		
1	C2410C1	Data Structures and Algorithms-2	4	0	2	6	5
2	C2430C1	Cryptography & Network Security	4	0	2	6	5
3	C2410C3	Operating Systems	4	0	2	6	5
4	C2410C4	Computer Networking	4	0	2	6	5
		Total	**16**	**0**	**8**	**24**	**20**
		Project-Based Learning					
5	C2430A1	PBL-4.1: Firewalls: Configuration and Operations	0	2	0	2	2
6	C2430A2	PBL-4.2: WireShark: Installation, Configuration and Operations	0	2	0	2	2
		Total	**16**	**4**	**8**	**28**	**24**

Semester 5

Sr#	**Course Code**	**Course Name**	**Teaching Scheme Per Week (in Hrs.)**			**Hours**	**Credit**
			L	**T**	**P**		
1	C2510C3	Mobile Application Development	4	0	2	6	5
2	C2530C1	Enterprise IT Security	4	0	2	6	5
3	C2530C2	Introduction to Cyber Security & Ethical Hacking	3	0	2	5	4
4		Elective-1	3	0	4	7	5
		TOTAL	**14**	**0**	**10**	**24**	**19**
		Project-Based Learning					
5	C2510A1	PBL 5.1 - Javascript Framework	0	2	0	2	2
6	C2530A1	PBL 5.2 - OWASP TOP - 10 Web Applications Pen. Testing	0	2	0	2	2

		Total	**14**	**4**	**10**	**28**	**23**
		Elective-1 List					
1	C2530D1	Machine Learning (Same as C2520C1)					
2	C2530D2	Advanced Web Technologies					
Semester 6							
Sr#	**Course Code**	**Course Name**	**Teaching Scheme Per Week (in Hrs.)**			**Hours**	**Credit**
			L	**T**	**P**		
1	C2630C1	UML & Agile Methodologies	4	0	2	6	5
2	C2630C2	Vulnerability Assessment and Penetration Testing	3	0	2	5	4
3	C2630C3	Mobile and Wireless Security	3	0	2	5	4
4		Elective-2	4	0	2	6	5
		Total	**14**	**0**	**8**	**22**	**18**
		Project-Based Learning					
5	C2630A1	PBL 6.1 - SNORT : Installation, Configuration and Operations	0	2	0	2	2
6	C2630A2	PBL 6.2 - SURRICATA : Installation, Configuration and Operations	0	2	0	2	2
7	C2630A3	PBL 6.3 - Wi-Fi and Bluetooth Security	0	2	0	2	2
		Total	**14**	**6**	**8**	**28**	**24**
		Elective-2 List					
1	C2630D1	Software Quality Assurance and Testing					

2	C2630D2	Design and Implementation of API					
3	C2630D3	Cyber Security Using AI					
		Semester 7					
Sr#	**Course Code**	**Course Name**	**Teaching Scheme Per Week (in Hrs.)**			**Hours**	**Credit**
			L	**T**	**P**		
1	C2730C1	Cyber Forensics	4	0	4	8	6
2	C2730C2	Secure Software Design and Development	4	0	4	8	6
3	C2730C3	Malware Analysis	3	0	2	5	4
4		Elective-3	3	0	2	5	4
		Total	**14**	**0**	**12**	**26**	20
		Project-Based Learning					
5	C2730A1	PBL 7.1 - Information Security - Laws and Standards	0	2	0	2	2
		Total	**14**	**2**	**12**	**28**	**22**
		Elective-3 List					
1	C2730D1	Cloud Security					
2	C2730D2	IoT Security					
		Semester 8					

Sr#	Course Code	Course Name	Teaching Scheme Per Week (in Hrs.)			Hours	Credit
			L	T	P		
1	C2830P1	Capstone Project	0	0	24	24	12
2	C2830P2	Project Reporting and Guidance	0	4	0	4	4
3	C2830A_	Project-Based Learning/MOOC (As per NEP)	0	4	0	4	4
4	C28X0A1	Environmental Studies	2	0	0	2	2
		Total	**2**	**8**	**24**	**34**	**22**

ITM (SLS) Baroda University
School of Engineering
Department of Computer Science and Engineering

Course Name :Programming in Python-1
Course Type: Core
Course code: C2110C1

Teaching Scheme			Credits	Examination Marks				Total Marks
L	**T**	**P**	**C**	**Theory Marks**		**Practical Marks**		
				External	**Internal**	**External**	**Internal**	
3	**0**	**4**	**5**	**40**	**60**	**20**	**30**	**150**

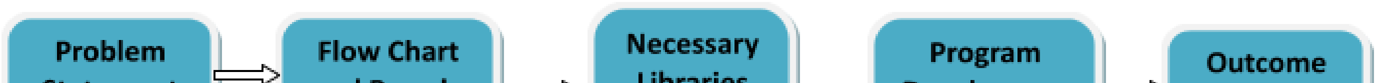
What is Python?

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Course overview

In this course, students will learn how to do programming in Python. Students will learn: how Python works and its place in the world of programming languages; to work with and manipulate strings; to perform math operations; to work with Python sequences; to collect user input and output results; flow control processing; to write to, and read from, files; to write functions; to handle exception; and work with dates and times.



Problem

Flow Chart

Necessary

Program

Outcome

Prerequisite

This course does not require any programming background. This course helps the students to learn programming in python.

Learning outcomes:

After completing the course, the student shall be able to :

	Course Outcome	Bloom’s Level
CO1	Understand basics concepts of Python programming	Understanding
CO2	Describe the Numbers, Math functions, Strings, List, Tuples and Dictionaries in Python	Application
CO3	Express different decision making statements and functions	Applying
CO4	Learn implementation essentials of python	Creation
CO5	Implement the specified mini-projects	Creation

Textbook:

Sr.No	Book Name
1	SheetalTaneja, Naveen Kumar, “Python Programming: A Modular Approach”, Pearson (2019)
2	R. NageswaraRao, “Core Python Programming”, dreamtech

Reference Book:

Sr.No	Book Name
1	ReemaThareja, “Python Programming: Using Problem Solving Approach”, Oxford University Press (2017)

2	John V Guttag, “Introduction to Computation and Programming using Python with Application to Understanding Data”, PHI (2016)
3	Martin C Brown, “Python: The Complete Reference”, McGraw-Hill Education (2018)
4	YashavantKanetkar, “Let us Python”, BPB Publication (2019)
5	Stephen Klosterman, “Data Science Projects with Python: A case study approach to successful data science projects using Python, pandas, and scikit”, Packt Publishing (2019) Kindle edition

Required Software:

1. Python Version 2.7 or 3.6 https://www.python.org/downloads/
2. Google Colab: https://colab.research.google.com/notebooks/io.ipynb

Learning Resources:

TedEx Videos:

Sr. No	TEDx Video
T1	https://www.youtube.com/watch?v=ENWVRcMGDoU– How algorithms shape our world.- Kevin Slavin Kevin Slavin argues that we're living in a world designed for -- and increasingly controlled by -- algorithms. In this riveting talk from TEDGlobal, he shows how these complex computer programs determine espionage tactics, stock prices, movie scripts, and architecture. Slavin also warns that we are writing code we can't understand with implications we can't control.
T2	World Changing: Data Science and AI | Fred Blackburn | TEDxURL The lecture covered the key points like tremendous increase in data, real world examples of machine learning i.eAlexa, Robot Scientist, Healthcare industry, artwork and many more.Video also covered the racing trends of living and working with human intelligence and the learning pattern of human mind and machine

Other Videos:

Sr. No	About Video	Link	Topic
O1	Erik Demaine, Ronald Rivest, and SriniDevadas.	https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-spring-2 008/	Introduction to Algorithm

	6.006 Introduction to Algorithms. Spring 2008. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu. License: Creative Commons BY-NC-SA		
O2	Lecture by Professor Jerry Cain for Programming Paradigms (CS107) in the Stanford University Computer Science department.	https://www.youtube.com/watch?v=Ps8jOj7diA0&list=PLD28639E2FFC4B86A	Programming Paradigm
O3	Dr. Anna Bell (MIT 6.0001 Introduction to Computer Science and Programming in Python, Fall 2016)	https://www.youtube.com/watch?v=RvRKT-jXvko&list=PLUl4u3cNGP63WbdFxL8giv4yhgdMGaZNA&index=17	List, Tuple and Dictionary
O4	Introduction to Python by Harvard University (Lecture-06 CS50 2018)	https://www.youtube.com/watch?v=mvlTSMUNQN4&t=1243s	Basics, Data Types, Control Statements

Related MOOCs courses:

Sr.No	**MOOC Courses**
M1	"The Joy of Computing Using Python" by Prof. SudarshanIyengar, IIT Ropar 12 Weeks on NPTEL.
M2	"Programming for Everybody (Getting Started with Python)",7 week course offered by University of Michigan and Courseera.

Course Outline:

Unit #	**Topics**	**Lab**	**Assignment**	**Teaching Hours**

1	**Introduction to Programming:** Introduction to Programming Fundamentals, programming environment, principles of programming, what is debugging, text editors and debuggers, introduction to Flow-Chart and Algorithm	P1	--	6
2	**Introduction to Python:** History, Features, Versions, Applications, Setting up path, Installation and Working with Python, Fundamentals of Python, Basic Syntax, Understanding Python variables, rules for naming identifiers and variables in python, operators and expressions, print(), type() and id() functions, taking user input using input() and raw_input() functions. **Data Types:** Integer, Float, complex numbers, Concept of Mutable and Immutable, String Manipulation.	P1	A1	8
3	**Data Types:** Lists, Tuples, Sets, Dictionaries, working with data types and their in-built functions, Logical Constructs, Boolean expressions, Looping, Python for loop, Python range(), Python Nested Loop Structures, Iteration, If-else, while loops, Break-Continue, Pass.	P2,P3, P4,P5, P6	A1,A2	8
4	**Functions and Recursion:** Defining a function, calling a function, Types of functions, Function Arguments, Anonymous functions, Passing Collections to a Function, Keyword and Optional Parameters, Local and global variables, Defining recursion and its application, programming through recursion	P7,P8, P9,P10	A1,A3	8
5	**Modules:** Importing Module, The Math Library, Random Numbers, Sys Module, OS, Date and Time module with their in-built functions	P11	A3	8
6	**File Handling:** File creation, open() and close() methods, read() and write() methods, file modes, file encoding, file object attributes, renaming and deleting files, Knowing Whether a File Exists or Not, Working with Binary Files, Appending Text to a File, Reading Text Files, File Exceptions, The with Statement, Python directory, directory methods and functions.	P12, P13	--	8
	TOTAL			**46**

Lab Experiments:

Sr. No.	Program Statement
P1	1. Write a python program to calculate the addition of two numbers without using third variable. 2. Demonstrate the use of id(), type() and size() function in python. 3. Write a program to perform basic operations in python.
P2	1. Write a program to find greatest among three numbers entered by user. 2. Write python code to check whether the entered number by user is even or odd. 3. Write a program to check whether the entered year is leap year or not. 4. Write a Python program to find all prime numbers within a given range.
P3	Write a program to take marks of 5 subjects from user and print obtained grade. (>=90% - A+, 70%-90% - A, 60%-70% - B+, 50%-60% - B, 35%-50% - C, <35% - F)
P4	Write a program to implement a simple calculator.
P5	1.Write a program for various functions of string in python. 2. To add 'ing' at the end of a given string (length should be at least 3). If the given string already ends with 'ing' then add 'ly' instead. If the string length of the given string 5. is less than 3, leave it unchanged. Sample String : 'abc' Expected Result : 'abcing' Sample String : 'string' Expected Result : 'stringly'
P6	1. Write a program for list and its various in-built methods. 2. Write a program to calculate the sum of numbers stored in a list. 3. Write a program for tuple and its various in-built methods. 4. Write a program for dictionary and its various in-built methods. 5. Write a program for set and its various in-built methods.
P7	Write a function that takes a list of numbers as input from user and produces the corresponding cumulative list where each element at index i is the sum of elements at index j <= i. For example, Input List = [3, 5, 2, 7, 9, 4]; the Output List = [3, 8, 10, 17, 26, 30].

P8	Write a program that takes a sentence as input from the user and computes the frequency of each letter. Use a variable of dictionary type to maintain the count.
P9	Write a function that takes a number as an input argument and returns the corresponding text in words, for example, if input is 368, the function should return ‘Three’, ‘Six’, ‘Eight’. Use a dictionary for mapping digits to their string representation.
P10	1. Write a Python program to find factorial of a given number using recursion 2. Write a Python program to print ‘n terms of Fibonacci series using recursion.
P11	1. Write a program to perform various functions of math module. 2. Write a program to perform various functions of random module. 3. Write a program to perform various system related functions using sys and os module. 4. Write a program to retrieve the date and time related information of a system using python.
P12	1. Write a python program that reads a text file and changes the file by capitalizing each character of file. 2. Write a python program to append data to an existing file 'python.py'. Read data to be appended from the user. Then display the contents of entire file.
P13	1. Read a text file in Python and print no. of lines and no. of unique words. 2. Write a python program to read line by line from a given files file1 & file2 and write into file3.

Assignments:

Sr. No.	Assignment Name
A1	**Guess the Number** The Goal: This assignment also uses the random module in Python. The program will first randomly generate a number unknown to the user. The user needs to guess what that number is. (In other words, the user needs to be able to input information.) If the user's guess is wrong, the program should return some sort of indication as to how wrong (e.g. The number is too high or too low). If the user guesses correctly, a positive indication should appear. You'll need functions to check if the user input is an actual number, to see the difference between the inputted number and the randomly generated numbers, and to then compare the numbers. Concepts to keep in mind: Random function Variables Integers Input/Output Print While loops If/Else statement

A2	Perform the following task using dictionary for the campus drive of a company which required 75 minimum CGPI criteria for appearing in the interview: a) Create the dictionary for department of CSE, Automobile, Mechanical, Civil and Electrical with name as key and their current CGPI with values. b) Combine all the department detail into one common dictionary and named it as ITM. c) Display the total number of students of university who are eligible for the drive. d) Display the total number of students of university who are not eligible for the drive.
A3	**Problem: An advanced Math Learning Tool** The program will generate just one question for each run that generate five questions and after a student answer all five, report the number of correct answers. The program should also display the time spent on test and list all the questions. **Sample Input and Output:** What is 9-2? 7 You are correct What is 3-0? 3 You are correct What is 3-2? 1 You are correct What is 7-4? 4 You answer is wrong 7-4 should be 3 What is 7-5? 3 You answer is wrong 7-5 should be 2 Correct count is: 03 Test time is 1021 seconds

	9-2=7 Correct 3-0=3 Correct 3-2=1 Correct 7-4=4 Wrong 7-5=4 Wrong

Course Name : R Programming for Data
Course Type: Core
Course Code:C2110C2

Teaching Scheme			Credits	Examination Marks				Total Marks
L	T	P	C	Theory Marks		Practical Marks		
				External	Internal	External	Internal	
4	1	2	6	40	60	20	30	150

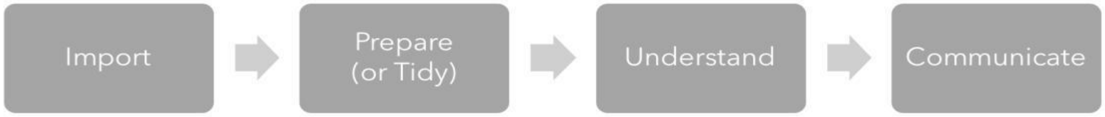
What is R?

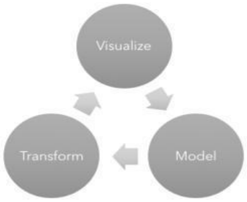
R is an open source language and environment for statistical computing, datamining, modeling, and data graphics. It provides a wide variety of statistical andgraphical techniques such as linear and non-linear modeling, statistical tests, timeseries analysis, classification, and clustering.

R is one of the most used business analytics tools. For example, Facebook uses R forbehavior analysis related to status updates and profile pictures. Google uses it toanalyze advertising effectiveness and economic forecasting. Twitter leverages R fordata visualization and semantic clustering. In the 2016 data science salary surveyconducted by O'Reilly, R was ranked second in a category of programminglanguages for data science.

Course overview

In this course, students will learn how to program in R and how to use R for effectivedata analysis and visualization. “Turn raw data into understanding, insight, andknowledge” (Wickham &Grolemund, 2017, p. ix) using R to import, prepare,understand, and communicate your data findings.





The course begins with developing a basic understanding of the R workingenvironment. Next, students will be introduced the necessary arithmetic and logicaloperators, salient functions for manipulating data, and getting help using R. Next,the common data structures, variables, and data types used in R will bedemonstrated and applied. Students will write R scripts and build R markdowndocuments to share their code others. They will utilize the various packagesavailable in R for visualization, reporting, data manipulation, and statistical analysis.

Students import data sets, transform and manipulate those datasets for variousanalytical purposes. Students will learn how to create control structures, such asloops and conditional statements to traverse, sort, merge, and evaluate data. Finally, students create interactive business applications that allow for data querying and data exploration.

Prerequisite

This course is designed for those who have no experience in R or programming. This class give you skills in programming in R and introduce you to 1) A new way ofthinking 2) A new language for speaking and reading (vectors, data frames, functions,objects, etc. and 3) a new syntax for writing , e.g. c(), print(), cat(), sort(), require(),subset() for data analysis and presentation.

Course outcomes

By the end of the course students you shall be confident and equipped with all the knowledge required to perform analytical activities in R. Specifically,

	Course Outcome	**Bloom’s Level**
CO1	**Understand** the fundamental syntax of R through readings, practice exercises, demonstrations, and writing R code.	Understanding
CO2	**Import** a variety of data formats into R using RStudio	Understanding
CO3	**Prepare** or tidy datas for in preparation for analysis	Applying
CO4	**Query** data using SQL and R	Application

CO5	**Analyze** a data set in R and present findings using the appropriate R packages	Analyze
CO6	**Visualize** data attributes using ggplot2 and other R packages.	Application

CO-PO Mapping

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PO13**	**PO14**	**PO15**	**PO16**
CO1	1	3		3	3	3	2	2		1	1	1	2	3	1	3
CO2	2	3	3	3	1	1	2	2	1				2	3	1	3
CO3	3	3	3	3	3	3	2	3	2	2	2	1	2	3	2	2
CO4	1	3	3	3	3	3	3	2	2		1			3	1	2
CO5	1	3	3	3	3	2	1	2	1			1	2	2		2
CO6		1	1	3	3	1	1	2			2			2		1

Tutorials, video demonstrations, and exercises

Sosulski, K. (2018). R Fundamentals.

Textbook

Sr. No	Book Name
1.	Wickham, H. &Grolemund, G. (2018). for Data Science. O'Reilly: New York
2.	Norman Matloff(2011),The Art of R Programming: A Tour of Statistical Software Design ,No Starch Press,

Reference Book

Sr. No	Book Name
1.	Jared P. Lander(2013), R for Everyone: Advanced Analytics and Graphics, Addison-Wesley Data & Analytics Series
2.	Mark Gardener(2013),Beginning R – The Statistical Programming Language, Wiley
3.	Robert Knell(2013),Introductory R: A Beginner's Guide to Data Visualisation, Statistical Analysis and Programming in R, Amazon Digital South Asia Services Inc,

Other Videos:

Sr. No	About Video	Link	Topic
O1.	Stanford Seminar - Expressing yourself in R,Hadley Wickham Rice University	https://www.youtube.com/watch?v=wki0BqlztCo	Data manipulation in R
O2.	MIT 15.071 The Analytics Edge, Spring 2017,MIT penCourseWare	https://www.youtube.com/watch?v=d2CfWJkklvo	Working with Data:History of R
O3.	MIT 15.071 The Analytics Edge, Spring 2017,MIT penCourseWare	https://www.youtube.com/watch?v=E_KUHMuoPLE	Getting Started in R

Related MOOCs courses

Sr.No	MOOC Course
M1	Introduction to R Software by Prof. Prof. Shalabh 8 Weeks on NPTEL.
M2	Data Science: Foundations using R, Johns Hopkins University(Coursera)

Required software

- R: http://www.r-project.org/ **(FREE)**
- RStudio (additional libraries required): http://www.rstudio.com/ **(FREE)**

.

Learning resources

- R Project: http://www.r-project.org/
- RStudio (additional libraries required): http://www.rstudio.com
- Quick-R http://www.statmethods.net/
- Google's R Style Guide: http://google-styleguide.googlecode.com/svn/trunk/Rguide.xml

Course Outline

Unit No.	Topics	Lab	Assignment	Hours
1	**Introduction to R programming** • What is R? • Installing R and RStudio • RStudio Overview • Working in the Console • Arithmetic Operators • Logical Operations • Using Functions • Getting Help in R and Quitting RStudio	P1	A1	4
2	**Data structures, variables, and data types** • Creating Variables • Numeric, Character and Logical Data • Vectors • Data Frames • Factors • Sorting Numeric, Character, and Factor Vectors • Special Values	P2	A2	4
3	**R packages and scripts** • Installing and loading packages • Setting up your working directory • Downloading and importing data	P3	A3	4

	• Working with missing data • Extracting a subset of a data frame • Writing R scripts • Adding comments and documentation • Creating reports			
4	**Descriptive statistics in R** • Measures of central tendency • Measures of variability • Skewness and kurtosis • Summary functions, describe functions, and descriptive statistics by group • Correlations	P4	A4	5
5	**Statistical graphs** • Scatter Plots • Box Plots • Scatter Plots and Boxand-Whisker Plots Together • Histograms	P5	A5	2
6	**Working with messy data** • Messy Data • Renaming Columns (Variable Names) • Attaching / Detaching • Tabulating Data: Constructing Simple Frequency Tables • Ordering Factor Variables	P6	A6	4
7	**Iteration** • while loops • for loops	P7		4
8	**Conditional Statements** • If / else • Boolean logical operators	P8		2

9	**Writing functions Reporting** • Creating functions • Calling functions	P9	A7	5
10	**Data Exploration and Visualization** • Using the ggplot2package to visualizedata • Applying themes fromggthemes to refine andcustomize charts and graphs • Building data graphics for dynamic reporting	P10		3
11	**Data querying: SQL and R** • Writing SQL statements in R • Using the Select, From, Where, Is, Like, Order By, Limit, Max, Min SQL Functions	P 11		4
12	**Interactive reporting with Rmarkdown** • RMarkdown basics • Text formatting • Code chunks • YAML header • Preview of notebooks,presentations ,websites, and dashboards	P12	A8	4
			Total Hours	45

Lab Experiments:

Sr.No	**Practical Name**
P1	Compute the following: 1. (123 - 45) / 4 + 4 * (72 / 2.34 - 3) 2. (((20*3)-14)^3)

3. Absolute value of -88
4. Base 10 logarithm of 72
5. Round the square root of 50 to the fourth decimal
6. e^1.45 - 2.612
7.
   a. Assign a variable year_born to 1984
   b. Assign a variable year_current to 2014
   c. Assign a variable age and compute it
   d. Return True / False if person is eligible to vote in US (if age is greater than or equal to 18)
6. Given: formula for area of circle is pi*r2 Given: Area = 100
   a. Write statement to find r. (Hint: utilize "sqrt" and "pi" functions)
7. Given: went to lunch and pre-tax bill was $45.90
   a. Compute subtotal: add NYC tax of 8.875%
   b. Compute 15% tip on subtotal
   c. Compute 20% tip on subtotal
8.
   a. Assign a variable customers to 500
   b. Assign a variable pizza_price to $20
   c. Assign a variable todays_revenue (customers * pizza_price) and compute today's revenue
   d. Is today's revenue greater than yesterday's revenue of $7,000 and less than tomorrow's projected revenue of $11,000? Show the code that would answer the following question.

P2

Dataset: Winter Olympic Medals.

Download Link: **http://becomingvisual.com/rfundamentals/winter_olympic.csv**

Data Dictionary: Review the data dictionary for the Winter Olympic Medals data set.

Variable	**Description**
Rank	Rank in number of medals

NOC	Name of country
Gold	Number of gold medals
Silver	Number of silver medals
Bronze	Number of bronze medals
Total	Total number of medals
Region	Country Region

1. Getting to know the data
   a. Import the data
   b. View the data
   c. How many variables are in the data frame?
   d. What are the names of these variables?
   e. How many countries (rows) are in the data frame?
2. Printing data
   a. The first row of data
   b. The last row of data
   c. The first 5 rows of data
3. Creating vectors
   a. Create a vector called "country_medals" from data frame
   b. Create a vector called "gold" from data frame
   c. What type of variable is "gold"?
4. Create a new data frame that holds data from the region Asia
   a. Call the data frame "asia"
   b. How many rows and columns are in this data frame? [Hint: use dim()]
5. Create the data frame "total_medals"
   a. Create vector "country"
   b. Create vector "total_medal_ct"

c. Use cbind() to combine the two vectors

d. What is the type of object “total_medals”?

6. Vector data counts

   a. What are the different levels of data$Region? [Hint: use levels()]

   b. Are any of the other variables factor variables? [Hint: use str()]

7. Subsetting

   a. Create a data frame that holds countries that did not win any gold medals

8.

   a. Create a vector called test_scores with the following values 92, 75, 84, 94, 88, 89, 91

   b. Create a vector called students with the following values Jerry, Monica, Felix, James, April, Ruth, Tony

   c. Create a data frame with these two vectors

   d. It turns out that Monica’s test was regraded and was awarded five extra points - correct this in the data frame.

   e. Extract the students who got above or equal to 90%

   f. Sort all the students by their test score in descending order

P3

Dataset: Basketball data from March Madness

Download Link: **http://becomingvisual.com/rfundamentals/march_madness.csv**

Data Dictionary:

Variable	**Description**
Rank	Team Ranking
Previous	Previous Team Ranking
School	Name of the College or University
Conference	NCAA Conference (30 +)
Record	Overall Record

Neutral	Record with games in a neutral location
Home	Record with games at home
Non Div I	Record with non-divison 1 games

Write a R script to do the following:

1. Set working directory – Hint: setwd()
2. Import the csv file
3. View the file
4. Print number of rows and columns – Hint: dim()
5. Print columns names
6. Change column names to lower case so it is easier to use Hint: names(df_name) <- tolower(names(df_name))
7. Explore the variable types. – Hint: str()
8. How many different conferences are there?
9. Let's look at the difference in values of first two columns:
    a. Compute a new vector called "diff" and calculate the difference in rank and previous
    b. Print count and list of schools that changed 3 or more places Hint: create subset that satisfies criteria
10. Import the GDP dataset and compute the difference in GDP between 2007 and 2017 for each country.

    Download: http://becomingvisual.com/rfundamentals/gdp.csv

    a. Create a subset of countries that saw an increase of over one trillion dollars.

P4

Create a RMarkdown (.Rmd) document that answers and addresses the following requirements.

1. Getting to know the data:
    a. Import the data (http://becomingvisual.com/rfundamentals/winter_olympic.csv)
    b. View the data
    c. Look at column names
    d. Look at dimension of data (rows and columns)

	2. Data is currently sorted by Rank. Sort data by total medals and country. Assign sorted data to a new data frame. Call it sort_total. 3. Use describe() function to look at data. a. If function does not work, first import library: library(Hmisc) 4. Look at some statistics a. What is median of number of gold, silver, bronze and total medals? b. Also look at the mean and total number of G, S, B and T medals 5. More statistics a. For Gold, look at summary stats, including: IQR, min, max, mean, var, sd, skew b. Use summary() and describe(). (May need to install library(psych)) 6. More statistics - subset a. Redo above statistics, this time group by Region b. Which region won the highest mean total medals? c. How many countries are in this Geographic Region? d. How many countries are in the EUROPE group? e. What is the max number of medals won? What country won the max? 7. More statistics – correlations a. Explore correlations between Total medals and number of Gold and Bronze b. What is the correlation between Rank and Total medals? Is this expected or surprising? 8. Import the GDP dataset and compute the measures of central tendency for 2017. (Divide by a trillion, and use na.rm = TRUE when computing the measures.) a. Find the mean b. Find the median c. Find the range d. Find the quantile
P5	In an R Markdown document, complete the following with the movies.csv data. Download the data from http://becomingvisual.com/rfundamentals/movies.csv 1. Getting to know the data a. Import the data b. View the data c. Look at column names

	d. Look at dimension of data (rows and columns) 2. Scatterplots a. Do scatter plot of Tickets Sold and Gross (Is the trend expected?) b. Redo scatter plot, adjusting scales, divide by 1000 c. Redo scatter plot, adjusting scales, divide by 100,000 d. Redo scatter plot, adjusting scales, divide by 1,000,000 3. What is the correlation between tickets sold and sales? Is this expected? 4. Scatterplots with lines a. Do scatter plot with millions scale, add a regression line b. Add label to x and y axis, add plot title label 5. Other plots a. Do boxplot b. Do boxplot - horizontal c. Do histogram for type of films d. Do histogram of gross sales. How bins are shown by default? e. Do histogram of gross sales with 10 bins. f. Do histogram of ticket sales. Try different bin numbers. g. Do histogram of ticket sales (use millions unit). Add frequency count to top of bars. Add titles. h. Do barplot of genre 6. In a R Markdown document, produce plots that describe the GDP (http://becomingvisual.com/rfundamentals/gdp.csv)and Life Expectancy (: http://becomingvisual.com/rfundamentals/life_expectancy.csv) during 2016 You will need to create a new data frame with these columns. a. Create a scatter plot of GDP to Life Expectancy b. Create a histogram of GDP c. Create a box and whisper plot of Life Expectancy
P6	Create an RMarkdown document to complete the following: 1. Getting to know the data a. Import the data (http://becomingvisual.com/rfundamentals/summer_winter_olympics.csv) b. View the data

c. Look at column names

d. Look at dimension of data (rows and columns)

2. Dealing with Data

   a. Look at the column names and change names to more meaningful names.

   b. The data represent, in order:

      1. country
      2. number of summer games played, gold, silver, bronze, total,
      3. number of winter games played, gold, silver, bronze and total, total
      4. total (Winter + Summer) games, gold, silver, bronze, total

3. Summary

   a. Use table() to find frequency of total summer games played

   b. Explore the data with other variables

4. Graphs

   a. Do histogram of summer games (total)

   b. Do histogram of winter games (total)

   c. Put above two histograms on one page

   d. Do two histograms on one page: total summer, total winter medals won

   e. Is there a correlation between number of medals given out in winter and summer? (do plot)

   f. How about number of games each country competes in. Is there correlation between winter and summer?

   g. Look at distribution of each of the types of medals, by season (6 histograms on one page)

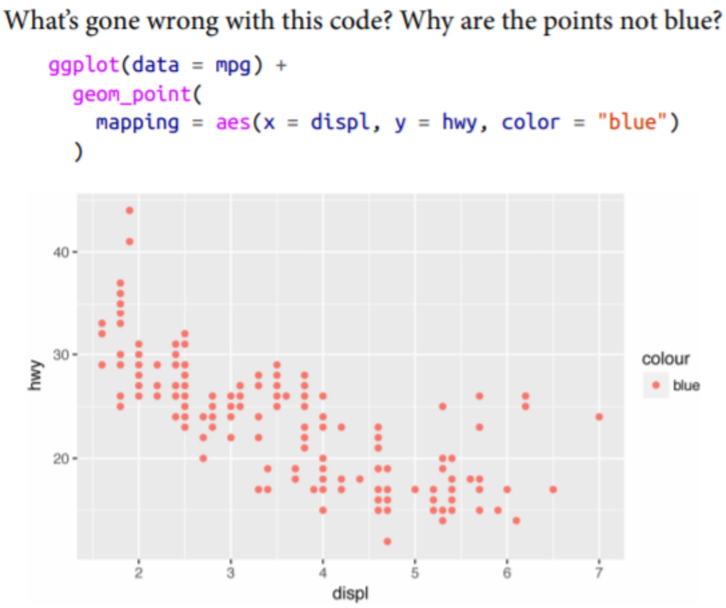
   h. Redo g with different number of bins (10 instead of 20)

   i. Explore data on your own

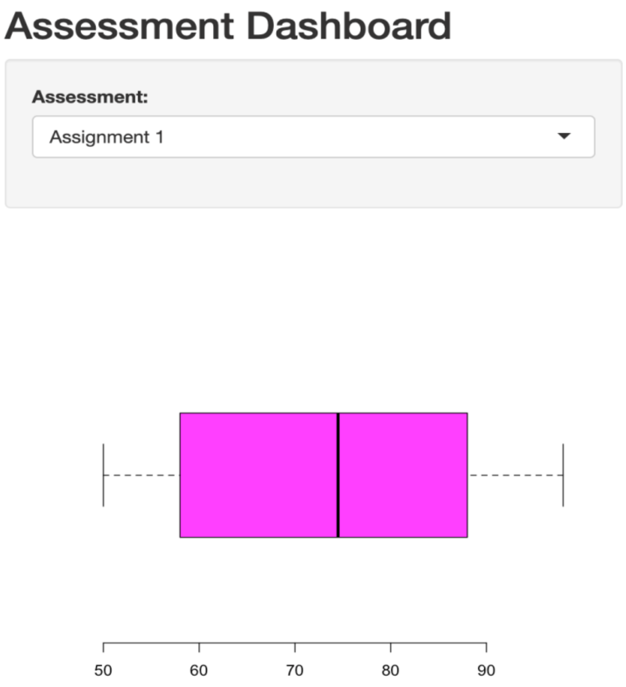
5. Merge the columns for the year 2016 for
   GDP (http://becomingvisual.com/rfundamentals/gdp.csv),
   Life Expectancy (http://becomingvisual.com/rfundamentals/life_expectancy.csv), and
   Employment (http://becomingvisual.com/rfundamentals/employment.csv) into a new
   data frame and clean-up the new table.

   a. Rename the appropriate columns to "country", "gdp", "life_expectancy", and "employment".

	b. Convert the employment number to percentages by dividing by 100 c. Then round life expectancy to zero decimals and employment to two decimals d. Create a frequency table for each variable e. Draw histograms for each variable
P7	Import the following AirBnb data set (http://becomingvisual.com/rfundamentals/airbnb.csv) Using a for loop count the number of AirBnbs that are in a particular neighbourhood the NYU area using the Greenwich Village and West Village neighbourhoods.
P8	Import the following AirBnb data set (http://becomingvisual.com/rfundamentals/airbnb.csv) Using if / else statement, count the number of AirBnbs that are in a particular neighbourhood the NYU area using the Greenwich Village and West Village neighbourhoods.
P9	1. Create a function that computes the mean, median, min, and max values. Use this function to compute those values for the attitude data set. 2. Create a new function called checkforna that checks to see if a give value is NA and prints out the row number and column name from the following data set: http://becomingvisual.com/rfundamentals/airbnb.csv
P10	Create a ggplot for following datasets: Data = MPG, Iris, Glass (From UCR Repository) 1. How many rows are in each dataset 2. How many columns are in each dataset? 3. Make a scatter plot of hwy vs cyL variables from mpg. 4.

	
	What's gone wrong with this code? Why are the points not blue?
	`ggplot(data = mpg) +`
	`  geom_point(`
	`    mapping = aes(x = displ, y = hwy, color = "blue")`
	`  )`
	 40
	30
	20
	hwy
	2
	3
	4
	5
	6
	7
	displ
	colour
	● blue

P11	SELECT all applicable data 1. The players on the San Antonio Spurs in 2014 2. Top 5 blockers in 2010 3. Top 10 combination power-forwards with the most defensive rebounds 4. Top 20 Player-seasons in the NBA 50-40-90 Club (players who have hit over 50% for FG%, 40% for 3P%, 90% for FT%, 300 field goals, 55 3-pointers, and 125 free throws) ordered by their amount of points Top 10 oldest Milwaukee Bucks players with over 1000 points.
P12	Using the http://becomingvisual.com/rfundamentals/nyuclasses.csv file, create a shiny app that displays a box plot of the student grades based on the assignment selection that looks like the image below:



Assessment Dashboard

Assessment:

Assignment 1

1. Revise the nyuclasses app to provide a default view of most recent distributions by most recent assignment due date.
2. Revise app to include a selector by one or more students.
3. Revise app to include doughnut charts to show completion, late or incompleted assessments by assessment type.

Assignments:

Sr. No.	**Assignment Name**
A1	a. Which of the following is a logical operator? / | - ^ b. What value does R return in the statement below? 3 >= 4 c. What is the result of this calculation? (45 + 3) * 43 + 3^2 d. How would R evaluate the following? carspeed = 70 speedlimit = 65 carspeed>speedlimit e. How would R evaluate the following? (2+2 == 4) | (2+2 == 5) f. How would R evaluate the following?

!FALSE

g. What is the result of this function?

round(33.2321435452, 2)

h. What is the result of this function?

sqrt (64)

i. What is the result of this statement?

sqrt(64) == 64 ^.5

j. What is the result of this statement?

abs(-32)

k. Which of the following is an arithmetic operator?

*, |, &, !

l. What is wrong with this code?

2 +   3 *4 + sqrt[100]

A2

Create a vector called unemploy_rate with 12 values, one for each month in 2013. The values for each month are listed below (beginning with January's rate of 7.9)

7.9	7.7	7.5	7.5	7.5	7.5	7.3	7.2	7.2	7.2

Create a vector called month and add 12 values, one for the name of each month in a year.

Jan	Feb	Mar	Apr	May	Jun	July	Aug	Sep	Oct

Convert month to a factor variable

Create a data frame called monthly_rate that is comprised of unemploy_rate and month.

How would you extract the unemployment rate for March?

Extract only those months where unemployment was below 7.5%.

What is a factor variable? When would you want to use a factor variable?

What is unique about a numeric variable?

Why would you use a data frame over a vector to store your data?

A3

Create a new R Script in RStudio named fed_stimulus.R

Add comments to your script that include your name and date

Go to NYC Open Data and export the Federal Stimulus dataset as a CSV file from https://data.cityofnewyork.us/Business/Federal-Stimulus-Data/ivix-m77e

Review the details of the variables included in the dataset by selecting the manage button on the NYC Open Data site for the Federal Stimulus data.

	Move the Federal_Stimulus_Data.csv file to your mydata folder on your desktop Import the dataset in RStudio. Change the name of the data frame from Federal_Stimulus_Data to fed_stimulus Compute the sum and mean for the payment value column Create a subset of your data that returns those projects with project status is equal to the completed 50% or more. Do not include fully completed projects. Review your R Script and add appropriate explanatory comments Try creating a knitr report (you do not need to hand this in).
A4	For this assignment use a pre-loaded dataset in R named attitude. This is from a survey of the clerical employees of a large financial organization, the data are aggregated from the questionnaires of the approximately 35 employees for each of 30 (randomly selected) departments. The numbers give the percent proportion of favorable responses to seven questions in each department. attitude is already pre-loaded in R. To view it, type >View(attitude). Create an R script that computes the measures of central tendency and measures of variability and the relationships for each of the seven variables in the attitude dataset. Use the functions below: mean, median, mode, max, min, range, quantile, IQR, var( ), sd( ), and cor( ) Check your work by using the summary and/or describe functions.
A5	Return to the attitude dataset. Produce at least one scatter plot, histogram, and box-and-whisker plot for each variable. Complete this as a R Markdown document. Optional: To save time, explore creating a matrix of histograms, a matrix of scatter plots, and a matrix of boxplots.
A6	Use the undersgraduate survey data from http://becomingvisual.com/rfundamentals/undergrad.csv to create ordered factor variables for the excel, statistics and programming variables. In a R Markdown draw histograms for your new ordered factor variables.
A7	Import the following AirBnb data set (http://becomingvisual.com/rfundamentals/airbnb.csv) 1. Create a set of functions that compute specific metrics by neighborhood: a. average_number_of_reviews b. average_price

	2. Use the functions created in part 1 to compute these metrics by room type and neighborhood.
A8	1. Identify your own data set and create a shiny app that allows the user to explore it (similar to the movie explorer). 2. Host the app on the shinyapps.io website. 3. Submit the URL to your published app.

ITM SLS
BARODA UNIVERSITY
"Think Big...
Think Beyond"

Course Type:

Course Code: S2110C1

SYLLABUS

Teaching Scheme			Credits	Examination Marks				Total Marks
L	T	P	C	Theory Marks		Practical Marks		
				External	Internal	External	Internal	
4	0	2	5	40	60	20	30	150

What is Discrete Mathematics?

Discrete Mathematics is a branch of mathematics involving discrete elements that uses algebra and arithmetic. It is increasingly being applied in the practical fields of mathematics and computer science. It is a very good tool for improving reasoning and problem-solving capabilities. This course explains the fundamental concepts of Sets, Relations and Functions, Mathematical Logic, Counting Theory, Probability, Mathematical Induction and Recurrence Relations, Graph Theory, Trees and Boolean Algebra.

What is Python?

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.

Python supports modules and packages, which encourages program modularity and code reuse.

The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Python has several built in libraries which can be used to understand the concept of Sets, Algebraic structures and Matrix representation of Graphs.

Course Overview

The purpose of this course is to understand and use (abstract) discrete structures that are backbones of computer science. In particular, this class is meant to introduce logic, proofs, sets, relations, functions, counting, and probability, with an emphasis on applications in computer science.

Students can also determine whether a mathematical argument is logically correct, study the relationship between finite sets, count the number of ways to arrange objects in a certain pattern and analyze processes that involve a finite number of steps.

Throughout the teaching of various topics in this subject, teachers discuss how a statement, a function or a complete Python program can help visualize and concretize that topic. It is a fond hope and desire of the designers of this course that many students will directly be helped by this and they will be better computer scientists/engineers having learned Discrete Mathematics with Python. Actually the Python language is separately taught in its full glory under a separate subject title.

Learning how to think mathematically is far more important than knowing how to do all the computations. Consequently, the principal objective of this course is to help you develop the analytic skills you need to learn mathematics. To achieve this goal, we will show you the motivation behind the ideas, explain the results, and dissect why some solution methods work while others do not.

This course will introduce you to the wonderful world of Python programming. You will learn about the essential elements of programming and how to construct basic Python programs for solving the examples in Discrete Mathematics. The topic becomes more understandable if the student is able to write a computer program in python and relate it with the manual solution. This will also develop the logical ability of the student. You will learn expressions, variables, functions, logic, and conditionals, which are used to solve the problems for most of the topic. You will also learn how to use Python modules, which enable you to benefit from the vast array of functionality that is already a part of the Python language. These concepts and skills will help you to begin to think like a computer programmer and to understand how to go about writing Python programs. By the end of the course, you will be able to write short Python programs that are able to accomplish real, practical tasks. This course is the foundation for understanding the python modules which are used in discrete mathematics for understanding the concepts of sets, objects, graphs, matrices which are fundamental for writing a program for several applications in real world life such as optimization problem, finding shortest path image processing etc.

Prerequisite: The students are required to have a reasonable mastery over Algebra, Logic.

Learning outcome-By the end of the course students shall be confident and equipped with elementary knowledge of Discrete Mathematics and how python is used for solving the problems.

	Course Outcome	**Bloom's Level**
CO 1	**Understand** the basic principles of sets and operations in sets and apply counting principles to determine probabilities, domain and range of a function, identify one-to-one functions, perform the composition of functions and apply the properties of functions to application problems.	Understanding
CO 2	**Write** an argument using logical notation and determine if the argument is or is not valid. To simplify and evaluate basic logic statements including compound statements, implications, inverses, converses, and contrapositives using truth tables and the properties of logic. To express a logic sentence in terms of predicates, quantifiers, and logical connectives.	Understanding
CO 3	**Apply** relations and to determine their properties. Be familiar with recurrence relations.	Applying
CO 4	**Interpret** different traversal methods for trees and graphs. Model problems in Computer Science using graphs.	Understanding
CO 5	**Analyze** how to write functions and pass arguments in Python.	Analyze
CO6	**Create** problems of Discrete mathematics and solve them using Python programs.	Creation

CO-PO Mapping

	PO 1	**PO 2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO 9**	**PO 10**	**PO 1 1**	**PO 1 2**	**PO 13**	**PO 1 4**	**PO 1 5**	**PO1 6**
CO1	1	2		3	3	3	2	2		1	1	2	2	3	1	3
CO2	1	3	3	3	2	1	2	2	1				2	3	1	3
CO3	3	3	3	3	3	3	2	3	2	2	2	1	2	1	2	2
CO4	1	3	3	3	3	3	3	2	1		1			3	2	3
CO5	2	3	3	3	3	2	1	2	2			1	2	3		2
CO6		2	1	3	3	2	2	2			2			2		1

Tutorials, video demonstration, and exercise

T1: Set theory, Function & Counting.
T2: Propositional Logic & Predicate Logic
T3: Relations, Partial ordering & Recurrence T4:
Graph theory
T5: Algebraic Structure
T6: Finite State Automata: Deterministic and Non Deterministic Finite State Automata.

https://youtu.be/XOH1wxrBMpE
Students will learn about sets, functions and relations.
https://youtu.be/mrCrjeqJv6U
Students will learn about the basic principles of counting.
https://youtu.be/E40r8DWgG40
Students will get good knowledge of graph theory.

Textbook

J. P. Tremblay and R. Manohar, Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw-Hill,1997
K. H. Rosen, Discrete Mathematics and its applications, Tata McGraw-Hill, 6th Ed., 2007.

Reference book

S. Lipschutz and M. L. Lipson, Schaum's Outline of Theory and Problems of Discrete Mathematics, 2nd Ed., Tata McGraw-Hill,1999.
David Liben-Nowell, Discrete Mathematics for Computer Science, Wiley publication, July 2017.
Eric Gossett, Discrete Mathematics with Proof, 2nd Edition,Wiley publication, July 2009.
Al Doerr and Ken Levasseur, "Applied Discrete Structures" 2020, 3rd Edition - version 7
https://faculty.uml.edu/klevasseur/ADS2

Required software

https://www.python.org/downloads/: Python Software.

Learning resources

https://swayam.gov.in/nd1_noc20_cs82/preview
Discrete Mathematics - By Prof. SudarshanIyengar, Prof, Neeldhara – IIT Ropar, IIT Gandhinagar.
https://www.edx.org/course/probability-the-science-of-uncertainty-and-data
Probability – The Science of Uncertainty and data by Massachusetts Institute of Technology.
https://youtu.be/h_9WjWENWV8 Course on Discrete Mathematics byMIT.
https://www.py4e.com/book.php
https://Py4e.com
https://colab.research.google.com/

Pedagogy:

The real value of Discrete Mathematics is in the abstract, mathematical models which help a Computer Scientist or Engineer to think clearly about complex systems, keeping away mundane details. However, study of such models requires the student to have background in more basic topics like Set theory, combinatorics, propositional logic, and basic matrix algebra. Thus the time spent in studying these basic topics will pay out well in the more abstract topics which are studied later on.

Another factor which helps the student is properly selected examples, both in terms of quality and quantity, that are relevant to the material being introduced.

With each major topic being covered is accompanied by at least one application in the computer field. One of the group of most immediate applications would be in present day complex computer software and that is where dovetailing a very convenient programming language like Python is justified. The teacher demonstrates in the class links to Python program constructs for a particular topic and the students later on develop and test the complete programs, which helps manifestation of abstract concepts in form of tangible results of a running program.

B.Tech (Discrete Mathematics with Python)		Pedagogy								
S.No.	Topic	Lecture	Workshop	ABL / PBL	Tutorial	Practical	Industrial Visit	Expert Lecture	MOOC/ NPTEL	Any other Tool
1	Set theory, Function & Counting	6	-	-	1	P-1, P-2, P3	-	-		
2	Propositional Logics, Predicate Logics	7	-	-	2	P-4, P-5, P-6	-	-		
3	Relations, Partial Ordering, Recurrence Relation	9	-	-	3	P-7, P-8, P-9	-	-		
4	Graphs, Types of Graph, Subgraph, Connectedness	9	-	-	4	P-10	-	-		
5	Algebraic Structures	4	-	-	5		-	-		
6	Deterministic Finite Automata, Nondeterministic Finite Automata	4			6	P-11				

Course Outline

Class	Topic		Reading s	La b	Tutorial
1	Set Theory	Definitions- Inclusion, Equality of Sets, Cartesian product, The Power Set, Some operations on Sets, Venn Diagrams Basic Concepts of Set Theory Some Basic Set Identities Understanding set operations using Python	Lesson 1	P-1	T1
2	Function	Introduction & definition Co-domain, range, image, value of a function, Examples Surjective, injective, bijective; examples Composition of functions, examples Inverse function, Identity map Condition of a function to be invertible, examples Inverse of composite functions Properties of Composition of functions Defining the mathematical expressions in python	Lesso n 2-3	P-1	T1
3	Counting	The Basics of Counting The Pigeonhole Principle Permutations and Combinations Binomial Coefficients Generalized Permutations and Combinations Generating Permutations and Combinations. Solving different problem related to Counting using python	Lesso n 4-6	P-2	T1

4	Propositio n al Logics	Definition, Statements & Notation Truth Values, Connectives Statement Formulas & Truth Tables Well-formed Formulas Tautologies Equivalence of Formulas Duality Law, Tautological Implications, Examples Formation of truth table using python.	Lesson 6-9	P-3, P-4	T2
5	Predicat e Logics	Definition of Predicates Statement functions, Variables, Quantifiers Predicate Formulas, Free & Bound Variables The Universe of Discourse, Examples Valid Formulas & Equivalences, Examples.	Lesso n 10-12		T2
6	Relations	Definition, Binary Relation Representation, Domain, Range, Universal Relation, Void Relation, Union, Intersection,	Lesson 13	P-8	T2

		and Complement Operations on Relations Properties of Binary Relations in a Set: Reflexive, Symmetric, Transitive, Anti- symmetric Relations Understanding the properties of Relations using python.			
7	Matrices & Equivalenc e Classes	Definition of Matrices Addition, subtraction, transpose Multiplication by a scalar, multiplication of two matrices, special matrices like Identity, determinant and inverse of a 2x2 matrix. Relation Matrix and Graph of a Relation Partition and Covering of a Set, Equivalence Relation, Equivalence Classes Compatibility Relation, Maximum Compatibility Block, Composite Relation Converse of a Relation, Transitive Closure of a Relation R in Set X . Performing the matrix operation using Python.	Lesson 14-16	P-5	
8	Partial Ordering	Definition, Examples Simple or Linear Ordering Totally Ordered Set (Chain), Frequently Used Partially Ordered Relations Representation of Partially Ordered Sets Hasse Diagrams, Least & Greatest Members, Minimal & Maximal Members, Least Upper Bound (Supremum), Greatest Lower Bound (infimum) Well-ordered Partially Ordered Sets (Posets) Lattice as Posets, complete Distributive modular and complemented lattices Boolean and pseudo Boolean lattices. Finding the Least, Maxima element from the POSET using python.	Lesson 17-19	P-6	T3

9	Recurrenc e Relation	Introduction, Recursion Recurrence Relation Solving, Recurrence Relation Solving the recurrence relations using python.	Lesso n 20-21	P-7, P-9	T3
10	Graph	Introduction, definition, examples Nodes, edges, adjacent nodes, directed and undirected edge Directed graph, undirected graph, examples Initiating and terminating nodes, Loop (sling) Distinct edges, Parallel edges	Lesson 22		T4

11	Types of Graph	Multi-graph, simple graph, weighted graphs, examples Isolated nodes Null graph; Isomorphic graphs, examples Degree, In-degree, out-degree, total degree of a node, examples	Lesso n 23-24		T4
12	Subgraph	Definition, examples Converse (reversal or directional dual) of a digraph, examples Path: Definition, Paths of a given graph, length of path, examples Simple path (edge simple), elementary path (node simple), examples Cycle (circuit), elementary cycle, examples	Lesson 25-26		T4
13	Connected ness	Definition, weakly connected, strongly connected, unilaterally connected, examples Strong, weak, and unilateral components of a graph, examples Matrix representation of graph: Definition, Adjacency matrix, Boolean (or bit) matrix, examples Determine number of paths of length n through Adjacency matrix, examples; Path (Reachability) matrix of a graph, examples Warshall's algorithm to produce Path matrix, Flowchart. Obtaining Boolean matrices, Adjacency matrix using python.	Lesson 27-30	P-10	T4
14	Algebrai c Structure s	Algebraic structures with one binary operation – semigroups Monoids and Groups congruence relation and quotient structures Free and cyclic monoids and groups permutation groups substructures Normal subgroups	Lesso n 31-34		T5

		Understanding Algebraic structures using Python			
15	Determinist ic Finite Automata	Alphabets, Strings, Languages Finite Automata (FA), acceptance of strings, and languages Deterministic Finite Automata (DFA)	Lesso n 35-36		T6
16	Nondeterm inistic Finite Automata	An informal view of Nondeterministic Finite Automata Definition of Nondeterministic Finite Automata, The extended transition function The language of an NFA, Equivalence of Deterministic and Nondeterministic Finite Automata.	Lesso n 37-38	P-11	T6

List of Program using Python for Discrete Mathematics:

P-1. Write a python program to define a Set, List the elements and perform the basic operations such as : Union, Intersection, Complement, Cartesian product of two sets.

P-2. Write a program to find the factorial of a number.

P-3. Write a program to find scalar multiply, matrix multiply of matrix.

P-4. Write a Python function that takes a list of pairs as an argument and determines if the argument constitutes a function or a general relation.

P-5. Write a program to denote Relation: as subsets of a power set, matrix representation.

P-6. Write a program for Propositional Logic, Logic operators, Truth Tables.

P-7. Write a program for Equivalence and implication.

P-8. Write a program for Functions: functional Python programming.

P-9. Write a program for Recursion: functions, generators.

P-10. Write a program for Graphs: matrix representation, association.

P-11 Write a program for Regular Expressions, Finite State Machine.

Web Technologies: HTML, CSS, JavaScript, PHP

Course Type:Core

Course code: C2110C3

Teaching Scheme			Credits	Examination Marks				Total Marks
L	T	P	C	Theory Marks		Practical Marks		
				External	Internal	External	Internal	
2	0	4	4	40	60	20	30	150

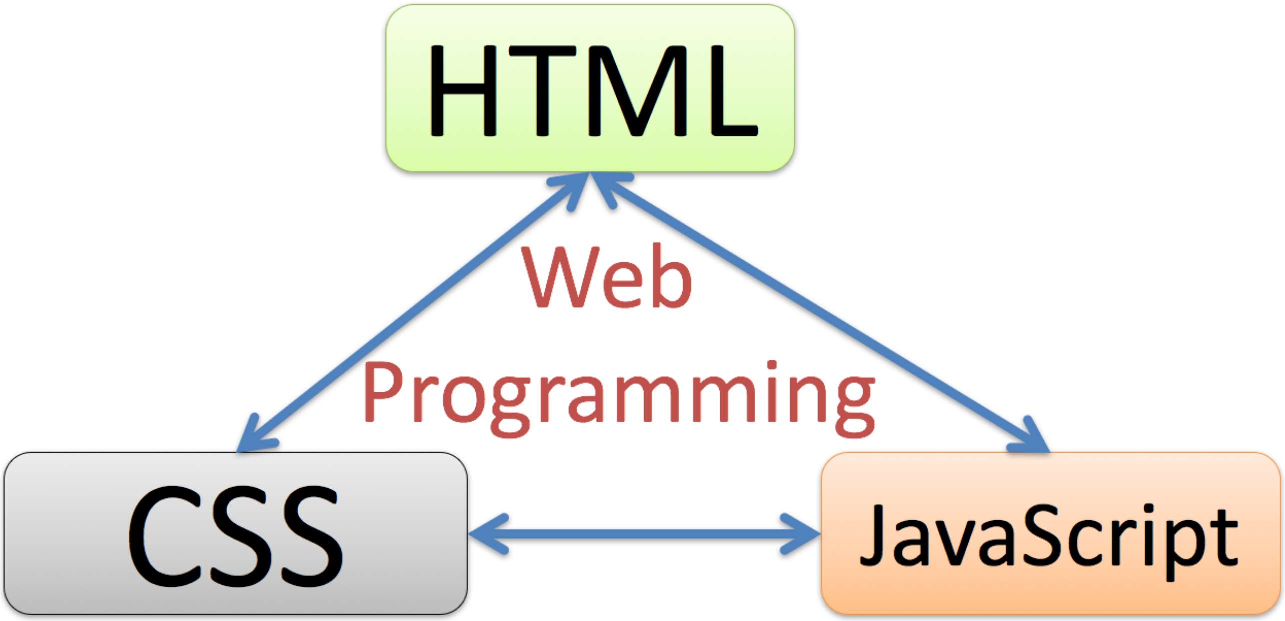
What is Web Technology?

Learning Web Technology is essential today because Internet has become the number one source to information, and many of the traditional software applications have become Web Applications. Web Applications have become more powerful and can fully replace desktop application in most situations.

That's why you need to know basic Web Programming, including HTML, CSS and JavaScript. It all started with Internet (1960s) and the World Wide Web - WWW (1991). The first Web Browser, Netscape, came in 1994. This was the beginning of a new era, where everything is connected on internet, the so-called Internet of Things (IoT).

Course overview

In this course students will start on a path toward future studies in web development and design, no matter how little experience or technical knowledge you currently have. The web is a very big place, and if you are the typical internet user, you probably visit several websites every day, whether for business, entertainment or education. But have you ever wondered how these websites actually work? How are they built? How do browsers, computers, and mobile devices interact with the web? What skills are necessary to build a website? With almost 1 billion websites now on the internet, the answers to these questions could be your first step toward a better understanding of the internet and developing a new set of internet skills.



By the end of this course you'll be able to describe the structure and functionality of the world wide web, create dynamic web pages using a combination of HTML, CSS, and JavaScript, apply essential programming language concepts when creating HTML forms, select an appropriate web hosting service, and publish your WebPages for the world to see. Finally, you'll be able to develop a working model for creating your own personal or business websites in the future and be fully prepared to take the next step in a more advanced web development.

Learning outcomes

By the end of the course students you shall be confident and equipped with all the knowledge required to design an effective web site

	Course Outcome	**Bloom's Level**
CO1	**Remember** the concepts of WWW including browser and HTTP protocol.	Understand
CO2	**Understand** various HTML Tags to develop user friendly web pages	Understand
CO3	Understand, analyze and build dynamic and interactive web sites	Understanding, Analyze
CO4	**Apply** CSS with its types to provide the styles to the web pages at various levels	Application

CO5	**Apply** JavaScript to develop the dynamic web pages	Analyze
CO6	Use server-side scripting with PHP to generate the web pages dynamically.	Application
CO7	Be able to write the schema for the given XML documents in both DTD and XML Schema languages	Creation
CO8	**Create** their own personal and professional websites.	Creation

TextBooks:

Sr. No	Book Name
1	Ralph Moseley and M. T. Savaliya, "Developing Web Applications", Wiley-India
2	Bai and Ekedhi, The Web Warrior Guide to Web Programming, 3rd Edition, Thomson, 2008
3	Web Technologies, Black Book, dreamtech Press
4	Knuckles, "Web Applications: Concepts and Real World Design", Wiley-India
5	Developing Web Applications in PHP and AJAX, Harwani, McGrawHill
6	Internet and World Wide Web How to program, P.J. Deitel & H.M. Deitel, Pearson
7	The complete reference PHP- Steven Holzner, Tata mcGraw-Hill

Required software

- Notepad++
- VS Code
- Codepen (Online Tool

Learning Resources

TEDx Videos:

Sr. No	TEDx Video
T1	https://www.ted.com/talks/kevin_kelly_how_technology_evolves?language=en#t-17539 How technology works.- Kevin Kelly Tech enthusiast Kevin Kelly asks "What does technology want?" and discovers that its movement toward ubiquity and complexity is much like the evolution of life.
T2	https://www.youtube.com/watch?v=Va07q3HFEZQ – The Dark Web – Alan Pearce Alan talks about the dark web and how it can be used for worthwhile purposes.

Other Videos

Sr. No	About Video	Link	Topic
O1	Stanford Lecture - Ben Galbraith: Web Skills: Introduction to Web Technologies and HTML 5	https://www.youtube.com/watch?v=r4F5tY_6u70&list=PLDD1386C4D784B56B&index=3&t=0s	Introduction to web
O2	James Mickenss. Securing Web Applications MIT OpenCourseWare	https://www.youtube.com/watch?v=WlmKwIe9z1Q	Securing Web Applications
O3	Prof.I.Sengupta, Department of Computer Science & Engineering ,IIT Kharagpur On HTML	https://www.youtube.com/watch?v=QEtWL4lWlL4	HTML

O4	Introductory lecture on Basic concepts of web applications, how they work and the HTTP protocol	https://www.youtube.com/watch?v=RsQ1tFLwldY	HTTP and WWW
O5	Douglas Crockford: The JavaScript Programming Language Douglas Crockford explores not only the language as it is today but also how the language came to be the way it is.	https://www.youtube.com/watch?v=v2ifWcnQs6M&list=PL62E185BB8577B63D	JavaScript
O6	Building Dynamic websites. Harvard OpencourseWare. CS E-75	https://www.youtube.com/watch?v=h2Nq0qv0K8M	How tom create dynamic website
O7	David Malan : PHP Harvard Web Development	https://www.youtube.com/watch?v=gwUEjgbs02w&list=PLTney8JqQkkk2oGFQONHHKDzvsChJMKEY&index=2	PHP

Related MOOCs courses

Sr. No.	MOOC Courses
M1	Introduction to HTML5 by University of Michigan 4 Weeks on Coursera.
M2	Advanced Styling with Responsive Design By University of Michigan on (coursera)
M3	https://www.edx.org/professional-certificate/w3cx-front-end-web-developer
M4	https://www.edx.org/course/javascript-introduction

M5	https://www.udemy.com/courses/development/web-development/
M6	https://www.edx.org/course/css-basics
M7	https://www.edx.org/course/programming-for-the-web-with-javascript

Course Outline:

Unit No	**Topic**	**Lab**	**Hours**
1	**Introduction to WWW** ● Concept of WWW ● Internet and WWW ● HTTP protocol: Request and Response ● Web Browser and Web Servers ● Features of Web 2.0 ● Setting up Unix and Linux Web Servers ● Dynamic IP ● Concepts of effective Web Design ● Web Design Issues	**-**	**04**
2.	HTML ● Basics of HTML ● Formatting and Fonts ● Color and Hyperlinks ● List, Table, Images, Forms ● HTML5 New Elements ● HTML5 : Drag/Drop, Video, Audio, Input types ● XHTML: Meta Tags	P1, P2, P3, P4, P5	**07**
3.	CSS ● Need for CSS ● Introduction to CSS ● Syntax and Structure ● Background images, Colors and properties ● Manipulation of Texts: Font, Border and Boxes ● CSS positioning ● CSS2, Overview and Features of CSS3	P1, P2,P3,P4,P5	**04**

4.	Introduction to JavaScript and JQuery ● Client side Scripting with JavaScript: Variables, conditions, Loops, Functions, Pop up Boxes ● Advance JavaScript : JavaScript and objects, ● DOM ● Forms and Validations ● DHTML: Combining HTML , CSS and JavaScript ● Events and Buttons ● AJAX: Introduction. ● AJAX Based WebApplications ● JQuery: Introduction	P6,P7,P8,P9,P10, P11,P16	**15**
5	XML ● Introduction to XML ● XML Key Components ● DTD and schemas ● Transforming XML using XSL and XSLT	P12, p13	**08**
6	PHP ● Introduction to Server Side Scripting ● Introduction and Basic Syntax of PHP ● String Processing and Regular expression ● Web Development Frameworks	P14, P15	**10**

Lab Experiments:

P1	**Creating a Food Drive Web Page** *Problem:* You did volunteer work for the Community Food Drive in your city. You would like to recruit other friends to volunteer for community service. You have been asked to create a Web page to display information about why you choose to volunteer and let people know how they also can help

P2.	**Creating a Web Page with Links:**Create a Web page demonstrating your knowledge of link targets
P3	*Problem:* Your Communications instructor has asked each student in the class to create a two-page Web site to help students in the class get to know more about the area in which you are majoring in school. The first Web page is a home page that includes basic information about your major. If you can, add an image related to your chosen field somewhere on the Web page. Add a link to the second Web page. The second Web page includes a paragraph of text and numbered lists with links.
P4	**Creating a School Bookstore Survey** *Problem:* The staff of the school bookstore wants to survey students about their book-buying habits to determine where they purchase their books. The staff has asked you to create a Web page form that contains the questions (create a form containing the basis information like name, address, city, zip, Email id ask the questions containing the survey.
P5	**Frames** **Creating a Two-Frame Structure for a Soccer Web Site** Problem: The Director of the Schererville Soccer league has asked you to create a new Web site with information on the upcoming soccer season. After reviewing the content with the Director, you suggest using a two-frame structure with two horizontal frames. The top frame will display a header and menu bar for navigation, while the bottom frame will display schedules, standings, rules, and other information.
P6	Write a JavaScript program for performing Arithmetic Operations
P7	Text growing and shrinking in Java script
P8	Design HTML form which includes two fields username and password. Write JavaScript code to show and hide password.
P9	Design a login form using HTML & JavaScript with following validations on username and password fields: Also, password length must be of 6 to 12 characters. Username should not start with _, @ or number and both the fields should not be blank.
P10	Write JS to demonstrate various built-in string functions Develop and demonstrate a HTML5 file that includes JavaScript script that uses functions for the following problems: a. Parameter: A string Output: The position in

	the string of the left-most vowel b. Parameter: A number Output: The number with its digits in the reverse order
P11	Write JS to search an element in an array of size N.
P12	Create an XML document that contains 10 users information. Write a java program, which takes user id as input and returns the user details by taking the user information from XML document using a)DOM parser b)SAX parser
P13	Introduction **XSLT**: Prepare a XSL file to transform Students data from XML to tabular form in XHTML.
P14	Write a PHP program named states.py that declares a variable states with value "Mississippi Alabama Texas Massachusetts Kansas". write a PHP program that does the following: a. Search for a word in variable states that ends in xas. Store this word in element 0 of a list named statesList. b. Search for a word in states that begins with k and ends in s. Perform a caseinsensitive comparison. [Note: Passing re.Ias a second parameter to method compile performs a case-insensitive comparison.] Store this word in element1 of statesList. c. Search for a word in states that begins with M and ends in s. Store this word in element 2 of the list. d. Search for a word in states that ends in a. Store this word in element 3 of the list.
P15	Create a XHTML form with Name, Address Line 1, Address Line 2, and E-mail text fields.
P16	Using AJAX retrieve data from a TXT file and display it. Create XSL file to convert XML file into XHTML file

ITM (SLS) Baroda University
School of Computer Science, Engineering and Technology
B.Tech - Semester II

Course Name: Programming in C
Course Code: C2210C1
Course Type: Core

Teaching Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
L	T	P	C	Theory Marks		Practical Marks		
				External	Internal	External	Internal	
3	0	4	5	40	60	20	30	150

Preamble:
For a given problem, developing logic, translating that logic into a programming language statements and implementation in a programming language is the skill required by programmers. Basic constructs of most of the programming languages are common. Under this backdrop and having gone through one programming language (Python-I during Semester I), the orientation in this course will be to learn the skills to develop logic for a given problem, select proper C language constructs and data elements and structures required to express that logic and its implementation in C Programming Language.

What is C?
C is a general-purpose programming language that is extremely popular, simple, and flexible to use. It is a structured programming language that is machine-independent and extensively used to write various applications, Operating Systems like Windows, and many other complex programs like Oracle database, Git, Python interpreter, and more. C was originally developed at Bell Labs by Dennis Ritchie between 1972 and 1973.

Many later languages have borrowed syntax/features directly or indirectly from C language. Like syntax of Java, PHP, JavaScript, and many other languages are mainly based on C language.

Course overview:
In this course, students will learn how to do programming in C. Students will learn: how C works and its place in the world of programming languages; to work with and manipulate strings; to perform math operations; flow control processing in C; to write functions; to work

with pointers and arrays in C.

Flowchart
Write
Psuedcode
Program **Outcome**
Development

Understand the Problem

Prerequisite:
Draw

This course assumes a basic understanding of the principles of programming, python programming-1 course in semester I.

Learning outcomes:
After completing the course, the student shall be able to:

	Course Outcome	**Bloom's Level**
CO1	Formulate algorithm/flowchart for given arithmetic and logical problem.	Understanding
CO2	Translate algorithm/flowchart into C program using correct syntax and execute it.	Understanding
CO3	Write programs using conditional, branching, iteration, and recursion.	Applying
CO4	Decompose a problem into function	Application
CO5	Develop an application using the concepts of array, and pointer to solve engineering and/or scientific problems.	Application

Programme Outcomes:

	POs	**Bloom's Level**
PO1	Design, develop and maintain computing systems using concepts from mathematics, science and engineering and program core courses	Creation
PO2	To implement software design systems, components, or processes to meet the desired needs and specifications.	Creation
PO3	Design algorithms for real world computational problems and analyze their complexities	Creation

PO4	To adapt to the usage of modern recent software tools and technologies and	Application
PO5	Analyze and interpret data and discover knowledge to provide solutions to engineering problems	Analyze
PO6	Assess the security, privacy, quality and cost parameters in developing software systems	Evaluate
PO7	Ability to work effectively in multi-disciplinary teams using common tools and standards to achieve project objectives	Application
PO8	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.	Creation
PO9	Proficiency in oral, written and visual means for technical presentations and documentations	Application
PO10	Engage in lifelong learning through self-directed and independent study of new techniques and tools	Creation

CO-PO Mapping:

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

Course Outline:

Unit #	**Topics**	**Lab**	**Assignment**	**Teaching Hours**

1	**Fundamentals of C:** Features of C language, structure of C Program, comments, header files, data types, constants and variables, operators (Arithmetic, relational, logical, increment, decrement, bitwise, assignment) expressions, conditional expressions, evaluation of expressions, type conversion, Flowchart and algorithm design. Overview of all concepts using C.	P1,P2, P3	A1	8
2	**Control structure in C:** Simple statements, Decision making statements (if-else, switch), looping statements (for, while, do-while), Nesting of control structures, break and continue, goto statement.	P4,P5, P6	A2	8
3	**Functions and Recursion:** Concepts of user defined functions, prototypes, definition of function, parameters, parameter passing, calling a function, scope and lifetime of variable, Recursion, as a different way of solving problems. Example programs, such as Finding Factorial, Fibonacci series	P7,P8	A3	8
4	**Pointers and Arrays:** Basics of pointers, pointer to pointer, pointer and array, pointer to array, array to pointer, function returning pointer, Concepts of array, one- and two-dimensional arrays, declaration and initialization of arrays, string, string storage, Initialization of pointer, pointers vs multidimensional arrays, command line arguments, pointers to functions	P9,P10,P11	A4,A5,A6	12
5	**Structures:** Basics of structures, structures and functions, arrays of structures, pointers to structures, self-referential structures, table lookup, typedef, unions, bit-fields	P12,P13		8
6	**Input and output:** standard input and output, formatted output, variable length argument list, formatted input, file access, Error handling-stderr and exit, line input and output, miscellaneous functions.	P14,P15	A7	8
7	**The UNIX System interface:** File Descriptor, Low level I/O- Read and Write, open, create, close, unlink, random access using Lseek, Fopen and Getc, Listing Directories, A storage Allocator	P16		4
8	**Standard libraries:** string.h, math.h, stdlib.h, assert.h, time.h	P17		2
	TOTAL			**58**

Textbooks:

Sr. No.	Book Name
1.	Brian W. Kernigham, Dennis Ritchie, “The C Programming Language”, Pearson
2.	R. G. Dromey, “How to Solve it by Computer”, Pearson

Reference Books:

1.	YeshvantKanetkar, “Let Us C”, BPB Publication
2.	Reema Thareja, “Computer Fundamentals and Programming in C”, Oxford Publication
3.	Herbert Schildt, “C: The Complete Reference”, McGrawHill

Required Software:

1. GNU C Compiler: https://gcc.gnu.org/
2. For working with embedded system corresponding architecture tool chain : Assembler, Link editor, etc.

Learning Resources:

1. https://ocw.mit.edu/courses/
2. https://www.edx.org/learn/c-programming

TedEx Videos:

Sr. No	TEDx Video
T1	**The Right Way to Learn Code l Brian Underwood, Max Ptacek& Ben Makarechian l TEDxYouth@SHC** **https://www.youtube.com/watch?v=PdTEXWtd3HY** Ben Makarechian, Brian Underwood, and Max Ptacek tell the story of their experiences trying to create a virtual reality dodgeball game with little knowledge of computer code. They share the methods they used to teach themselves coding so students around the world can learn on their own.
T2	**How algorithms shape our world. -Kevin Slavin** **https://www.youtube.com/watch?v=ENWVRcMGDoU** Kevin Slavin argues that we're living in a world designed for --and increasingly controlled by --algorithms. In this riveting talk from TEDGlobal, he shows how these complex computer programs determine espionage tactics, stock prices, movie scripts, and architecture. Slavin also warns that we are writing code we can't understand with implications we can't control.
T3	**Don't Just Learn to Code, Learn to Create l Justin Richards l TEDxYouth@ColumbiaSC**

	https://youtu.be/6rxWc-TNIJI Students around the world are told they must 'Learn to Code!' This certainly sounds great, but why should you learn to code? Will it really help you accomplish your personal goals or land your dream job? We'll see how learning to code is only a part of joining the digital revolution, but we'll go further to explore how you can harness technology to your unique way of impacting the world. Learn how you can shape the future by not just learning to code, but learning to create.

Other Videos:

Sr. No	**About Video**	**Link**	**Topic**
O1	Erik Demaine, Ronald Rivest, and Srini Devadas. 6.006 Introduction to Algorithms. Spring 2008. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu. License: Creative Commons BY-NC SA	https://ocw.mit.edu/courses/electrical engineering-and-computer-science/6-006-introduction-to-algorithms-spring-2008/	Introduction to Algorithm
O2	Lecture on what is computation; introduction to data types, operators, and variables by Prof. Eric Grimson, Prof. John Guttag	https://www.youtube.com/watch?v=k6Ui4gXkLM&list=PLo7g9OE1yqEKY4Kx AiPpqA9gp7XOcPQ2V	What is computation; introduction to data types, operators, and variables
O3	Lecture by Professor Jerry Cain for Programming Paradigms (CS107) in the Stanford University Computer Science department.	https://www.youtube.com/watch?v=Ps8jOj7diA0&list=PLD28639E2FFC4B86A	Programming Paradigm

O4	Lecture Series on Programming and Data Structure by Dr.P.P. Chakraborty, Department of Computer Science and Engineering, IIT Kharagpur	https://www.youtube.com/watch?v=5D5i erCsAkM&list=PLeCxvb23g7hrw27Xlek HtfygUTQ0TmFfP&index=9	Decompositio n using Recursion
O5	Lecture Series on Programming and Data Structure by Dr.P.P.Chakraborty, Department of Computer Science and Engineering, IIT Kharagpur	https://www.youtube.com/watch?v=3Xo6 P_V qns&list=PLeCxvb23g7hrw27XlekHtfyg UTQ0TmFfP&index=13	Array and Addresses

Related MOOCs courses:

Sr.No	**MOOC Courses**
M1	“Introduction to Programming in C” by Dr. Satyadev Nandakumar, IIT Kanpur 8 weeks on NPTEL.
M2	“Art of C Programming” by Dr. Lajisj V.L., Department of Computer Science, University of Calicut.

Lab Experiments:

Sr. No.	**Program Statement**
P1	1. Write a C program to print “Hello World…!!!” and find out what happens when printf’s argument string contains \c. 2. Write a C program to do following: a. Print corresponding Celsius to Fahrenheit table. b. Modify the program to print heading above the table. c. Print Celsius to Fahrenheit table in reverse order. d. Implement the program using user defined function for temperature conversion 3. Write a C program to print the value of EOF.
P2	1. Write a C program to copy input file to output file, replacing multiple blanks with a single one. 2. Write a C program to copy input file to output file, replacing each tab with ‘\t’ 3. Write a C program to print one word per line from the input file. 4. Write a C Program to print a histogram of the following from the input file a. lengths of words b. the frequencies of different characters in its input.

P3	1. Write a C program to print all input lines from the file that are longer than 80 characters. 2. Write a C program to remove all comments from a C program. Don't forget to handle quoted strings and character constants properly. 3. Write a C program to remove all trailing blanks and tabs from each line of input file, and to delete entirely blank lines.
P4	1. Write a C program to read marks of a student from the keyboard whether the student is pass or fail (using if else). 2. Write a C program to read three numbers from the keyboard and find the maximum out of these three. (nested if else). 3. Write a C program to check whether the entered character is capital, small letter, digit or any special character.
P5	1. Write a C program to read no 1 to 7 and print relatively day Sunday to Saturday (using switch case). 2. Write a C program to find out the Maximum and Minimum number from given 10 numbers.

P6	1. Write a C program to reverse a number. 2. Write a C program to input an integer number and check if the last digit of number is even or odd. 3. Write a C program to print the following pattern: i. * * * * * * * * * * ii. 1 2 3 4 5 6
P7	1. Write a C program to create a function named sum which calculates the sum of three given numbers. 2. Write a C program to convert decimal number to binary number using function. 3. Write a C program to find the prime numbers between given intervals using function.
P8	1. Write a C program to find the factorial of a given number. (recursion) 2. Write a C program to generate the first n number of Fibonacci series (recursion). 3. Write a C program to reverse a sentence using recursion.
P9	1. Write a C program to reverse an array using a pointer. 2. Write a C program to reverse string using pointer. 3. Write a C program to search an element in an array using a pointer.

P10	1. Write a C program to store n elements in an array and print the elements using pointer. 2. Write a C program to print all the permutations of the given string using pointer. 3. Write a C program having a pointer version of the function strcat(s,t) which copies the string t to the end of s.
P11	1. Write a C program for the versions of the library functions strncpy, strncat, and strncmp, which operate on at most the first n characters of their argument strings. For example, strncpy(s,t,n) copies at most n characters of t to s. 2. Write a C program expr , which evaluates a reverse Polish expression from the command line, where each operator or operand is a separate argument. For example, expr 2 3 4 + * evaluates 2 X (3+4).
P12	1. Write a C program to store information of N students using structure. 2. Write a C program to add two distances in inch-feet using structure. The values of the distance are to be taken from the user. 3. Write a C program to enter the marks of 5 students in Python, C and Discrete Mathematics (each out of 100) using a structure named Marks having elements roll no., name, py_marks, c_marks and dm_marks and then display the percentage of each student.
P13	1. Write a C program that prints the distinct words in its input sorted into decreasing order of frequency of occurrence. Precede each word by its count. 2. Write a C program to create a function undef that will remove a name and definition from the table maintained by lookup and install.
P14	1. Write a C program to write multiple lines in a text file. 2. Write a C program to read the file and store the lines into an array. 3. A file named data contains a series of integer numbers. Write a c program to read all numbers from file and then write all odd numbers into file named “odd” and write all even numbers into file named “even”. Display all the contents of these files on screen.
P15	1. Write a C program that converts upper case to lower or lower case to upper, depending on the name it is invoked with, as found in argv[0]. 2. Write a C program to print a set of files, starting each new one on a new page, with a title and a running page count for each file. 3. Write a C program to compare two files, printing the first line where they differ.
P16	1. Write a C program _flushbuf , fflush , and fclose. 2. The standard library function int fseek(FILE *fp, long offset, int origin) is identical to lseek except that fp is a file pointer instead of a file descriptor and the return value is an int status, not a position. Write a C program for fseek . Make sure that your fseek coordinates properly with the buffering done for the other functions of the library. 3. The standard library function calloc(n,size) returns a pointer to n objects of size size , with the storage initialized to zero. Write calloc , by calling malloc or by modifying it.

P17	1. Write a C program to perform various functions of string.h, math.h, stdlib.h, assert.h, time.h

Mini Project:

Apart from the above laboratory work, each student will implement a teacher specified individual small project requiring implementation through more than one source file and one header file, algorithm selection, determining the test data sets and a documentation for a 10-15 minutes presentation of his/her work. Some of the examples are listed below :

MP1	**Snake Game** Implement a snake game in which users can use up, down, right or left arrows to move the snake. There will be 3 lives and life will be decreased if you hit the snake's body or the wall. You have to provide food at several coordinates by which score and length of the snake will be increased.
MP2	**Bus Reservation System** Implement a system that can reserve seats on the bus. It can display the availability of seats and also can find the booking information.

Assignments:

Sr. No.	**Assignment Name**

A1	Write a C program which takes name, basic , daper (ie, percentage of D.A), bonper (ie, percentage bonus) and loandet (loan amount to be debited) for an employee. Calculate the salary using the following relation: salary = basic + basic * daper /100 +bonper * basic/100 - loandet Data is as following: **Name Basic daper bonper loadnet** Amit 2500 55 33.3 250.0 Calculate salary and then print the result under the following headings: **Name Basic Salary**
A2	Write a C program to generate a calendar for a given month and for a given year. (month and year is to be taken as input from user).

A3	Write 2 different C functions to compute the area and perimeter of a triangle whose sides a, b, and c are given by user as inputs. Formula to compute perimeter = a + b + c Formula to compute area = [s(s-a)(s-b)(s-c)]0.5 Where s = 0.5 * (a+b+c) Function prototypes are: double perim(double a, double b, double c) double area(double a, double b, double c) Your program should read the input data and print the output data via separate functions. The prototypes are: double read_input() double print_value(double val)
A4	Write a C program to perform the following operations on an integer array of 10 elements. Accept the values from the user. 1. Sort an array in ascending order. 2. Display sum of all odd values stored in an array. 3. Display number of even values stored in an array.
A5	Write a C program to implement an inventory system. Store the item number, name, rate and quantity on hand in a structure. Accept the details for five items into a structure array and display the item name and its total price. At the end, display the grand total value of the inventory.
A6	Write a C program that takes 2 integer sets A[] and b[] as input and prints results of following set operations: i. A union B (Write function set_union()) ii. A intersection B (Write function set_intersection()) iii. A-B and B-A (Write function set_difference()).
A7	Write a C program for the following: 1. A file name is command line argument. Display the contents of the file where each word will be displayed on a new line. Display a proper message if the file does not exist. 2. Display no. of vowels stored in the file. 3. Display no. of “the” stored in the file. 4. Copy contents of the file to another file.

Name	**Basic**	**daper**	**bonper**	**loadnet**	
Amit	2500	55	33.3	250.0	

Name	**Basic**	**Salary**	

ITM(SLS) Baroda University
School of Computer Science, Engineering and Technology
B.Tech - Semester II

Course Name : Data Structures and Algorithms-1
Course Code: C2210C2
Course Type: Core

Teaching Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
L	T	P	C	Theory Marks		Practical Marks		
3	0	4	5	External	Internal	External	Internal	
				40	60	20	30	150

Course Outcome:

After completing the course, the student shall be able to :

Sr. No.	Course Outcome	Bloom's Level
CO1	Understand Asymptotic Notation and concept of ADT	Understanding
CO2	Identify data structures suitable to solve problems	Understanding
CO3	Develop, analyze and implement algorithms for linear data structures like arrays and linked list	Applying
CO4	Develop, analyze and implement algorithms for non-linear algorithms like graphs and trees	Application
CO5	Implement searching and sorting algorithms along with Specified mini project.	Application
CO6	Understand the concept of Time Complexity using different sorting algorithms.	Application

Programme Outcomes:

Sr. No.	POs	Bloom’s Level
PO1	Design, develop and maintain computing systems using concepts from mathematics, science and engineering and program core courses	Creation
PO2	To implement software design systems, components, or processes to meet the desired needs and specifications.	Creation
PO3	Design algorithms for real world computational problems and analyze their complexities	Creation
PO4	To adapt to the usage of modern tools and technologies and recent software	Application
PO5	Analyze and interpret data and discover knowledge to provide solutions to engineering problems	Analyze
PO6	Assess the security, privacy, quality and cost parameters in developing software systems	Evaluate
PO7	Ability to work effectively in multi-disciplinary teams using common tools and standards to achieve project objectives	Application
PO8	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.	Creation
PO9	Proficiency in oral, written and visual means for technical presentations and documentations	Application
PO10	Engage in lifelong learning through self-directed and independent study of new techniques and tools	Creation

Competencies:

After completion of the this course, our students shall acquire the following generic competencies:

Sr. No	Competencies
1	To implement the searching techniques, for example, in stock market applications for buying and selling of different stocks.

2	To implement games based on Tree and Graph data structures, for example, the maze game using white block problem.
3	To implement applications involving sorting of structured data, with efficiency and stability requirements
4	To implement linked data structure based applications, for example, Implementations of polynomials operations using stack (Addition , subtraction)
5	To implement applications, using applicable Data Structures, starting with implementation of their ADT's, for example, the Queue ADT

Mapping:

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

Course Contents:

Unit	Topics	time(in hr)	% evaluation
1	**Introduction**: Pseudocode, Abstract Data Type, Data types – Primitive and non-primitive, Types of Data Structures- Linear & Non Linear Data Structures, ADT model and ADT implementation,Algorithm efficiency-Asymptotic Notations: Big O, Big Omega and Big Theta; Properties of Asymptotic Notations	6	5
2	**Recursion-** designing recursive algorithms,defining recursive algorithms,examples-factorial of a number, tower of hanoi , fibonacci series	4	5

3	**Linear Lists:** **Array:** Representation of arrays, Operations on Arrays, Applications of arrays **Stack:** Stack-Definitions & Concepts, Operations On Stacks, Applications of Stacks-Polish Expression, Reverse Polish Expression. **Queue:** Representation Of Queue, Operations On Queue, Circular Queue, Priority Queue, Array representation of Priority Queue, Double Ended Queue, Applications of Queue **Linked List:** Singly Linked List, Doubly Linked list,Circular linked list ,Linked implementation of Stack, Linked implementation of Queue, Applications of linked list	10	20
4	**Non-linear Lists:** **Tree**-Definitions,Representation of binary tree, Tree traversal -Inorder, postorder, preorder, Threaded binary tree, Binary search trees, Conversion of General Trees To Binary Trees.	6	20
5	Applications Of Trees Some balanced tree mechanism; e.g. Heap, AVL trees; 2-3 trees; Height Balanced, Weight Balance, Red black tree; Multi-way search tree: B and B+ tree;	5	15
6	**Graph:** Graph Terminologies, Adjacency Matrices and List Representations of Graphs; Elementary Graph Operations: Depth First Search & Breadth first Search, Spanning Trees: Shortest path, Minimal spanning tree using graphs,networks.	6	20
7	**Sorting**: basics, selection, insertion, exchange, QuickSort, external sort	4	10
8	**Searching**: Linear search, Binary search,list, hashed lists	4	5
Total Hours		**45 Hours**	

Reference Text:

Sr. No	Book
1	Forouzan and Gilberg ,"Data Structures - A pseudocode Approach with C" 2nd Ed.

Note: Python implementation of the various data structures and corresponding algorithms will be provided by the teacher. The students will be encouraged to study the C implementations given in the Reference Text (1).

Simulations/Animation:

TEDx Videos:

Sr. No	TEDx Video
T1	https://www.youtube.com/watch?v=ENWVRcMGDoU - How algorithms shape our world.- Kevin Slavin Kevin Slavin argues that we're living in a world designed for -- and increasingly controlled by -- algorithms. In this riveting talk from TEDGlobal, he shows how these complex computer programs determine espionage tactics, stock prices, movie scripts, and architecture. Slavin also warns that we are writing code we can't understand with implications we can't control.
T2	https://www.youtube.com/watch?v=6hfOvs8pY1k - What's an algorithm - David J. Malan An algorithm is a method of solving problems both big and small. Though computers run algorithms constantly, humans can also solve problems with algorithms. David J. Malan explains how algorithms can be used in seemingly simple situations and also complex ones.

Other Videos:

Sr. No	About Video	Link	Topic
O1.	Stanford Lecture - Don Knuth: The Analysis of Algorithms (2015, recreating 1969)	https://www.youtube.com/watch?v=v kU NH9r6UCI	Algorithm
O2.	Erik Demaine, Ronald Rivest, and SriniDevadas. *6.006 Introduction to Algorithms.* Spring 2008. Massachusetts	https://ocw.mit.edu/courses/electrical engineering-and-computer-science/ 6-006-introduction-to-algorithms-spring 2008/	Algorithm
	Institute of Technology: MIT OpenCourseWare,https://ocw.mi t.edu. License:Creative Commons BY-NC-SA.		

O3.	Dr. Rob Edwards from San Diego State University provides an introduction to complexity measures	https://www.youtube.com/watch?v=zgC nMvvw6Oo&list=PLpPXw4zFa0uKKh aSz87IowJnOTzh9tiBk	Stack , Queue , Linked list
O4.	MIT 6.0002 Introduction to Computational Thinking and Data Science, Fall 2016 View the complete course:[http://ocw.mit.edu/ 6- 0002F16](http://ocw.mit.edu/6-0002F16)Instructor: Eric Grimson	https://www.youtube.com/watch?v=B7h VxCmfPtM	Heap and Heaps sort
O5.	MIT 6.006 Introduction to Algorithms, Fall 2011 View the complete course:http://ocw.mit.edu/6-006F11Instructor: Victor Costan	https://www.youtube.com/watch?v=r5p Xu1PAUkI	Recursion Tree, Binary search tree
O6.	How to Construct a Tree | Stanford University	https://www.youtube.com/watch?v=QjF HWUsoZBw	Construct a Tree
O7.	Hashing-based data structures and applications - Michael Mitzenmacher, Harvard University	https://www.youtube.com/watch?v=g4Z yjRaYe3k	Hashing

Related MOOCs courses

Sr. No	MOOC Course
M1	Programming, Data Structures And Algorithms Using Python by Prof. MadhavanMukund 8 Weeks on NPTEL.
M2	Data Structures and Algorithms Specialization Offered By University of California San Diego National Research University Higher School of Economics(coursera)

Lab Experiments:

P1	Write a python code to demonstrate Call by Value and Call by reference.
P2	Write a python program to implement Bubble Sort and find the complexity.
P3	Write a python program to implement Linear search and Binary Search and find its time complexity.
P4	Implement a python program for stack that performs the following operations using array. (a) PUSH (b) POP (c) PEEP (d) CHANGE (e) DISPLAY
P5	Implement a python program to convert infix notation to postfix notation using stack.
P6	Write a python program to implement QUEUE using arrays that performs following operations (a) Insert (b) Delete (c) Display
P7	Write a python program to implement Circular Queue using arrays that perform the following operations. a. Insert b. Delete c. Display
P8	Write a menu driven python program to implement following operations on the singly linked list. (a) Insert a node at the front of the linked list. (b) Insert a node at the end of the linked list. (c) Insert a node such that the linked list is in ascending order(according to the info. Field) (d) Delete a first node of the linked list. (e) Delete a node before a specified position. (f) Delete a node after specified position.
P9	Write a python program to implement the following operations on the circular

	linked list. Insert a node at the end of the linked list. Insert a node before the specified position. Delete a first node of the linked list. Delete a node after specified position.
P10	Write a program python to implement following operations on the doubly linked list. (a) Insert a node at the front of the linked list. (b) Insert a node at the end of the linked list. (c) Delete a last node of the linked list. (d) Delete a node before a specified position.
P11	Write a python program which creates binary search trees and implements recursive methods for inorder,preorder and post order traversal.
P12	Write a python program to implement stack using linked lists.
P13	Write a python program to implement a queue using a linked list.
P14	Write a python program to implement Merge Sort
P15	Write a program python to implement Quick Sort
P16	Write a python program to implement DFS and BFS.
P17	Write a python program MID(Key,Hash)which uses the mid square method to find a 2-digit hash address HASH of 4 digit employee number key.

Mini Projects :

Apart from the above laboratory work, each student will implement a teacher specified individual small project requiring data structure selection, algorithm selection, determining the test data sets and a documentation for a 10-15 minutes presentation of his/her work. Some of the examples of projects are mentioned below.

Sr. No	**Project**
MP1	Design and Develop the index for a text book of at least 100 pages using alphabets.
MP2	Design a Student Prerequisite Subjects Management System requires the use of linked list or tree to store different courses and their prerequisites and based on this list it will allow any student to take any course or not.

MP3	Simulate a simple dictionary. Assume each character contains at least 10 vocabularies. Create an index page for all characters. Retrieve the word using index value. Assume that the index characters from a to z.

ITM(SLS) Baroda University
School of Computer Science, Engineering and Technology
B.Tech – Semester II

Course Name: Computer Graphics
Course Code:C2210C3
Course Type: Core

Teaching Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
L	T	P	C	Theory Marks		Practical Marks		
				External	Internal	External	Internal	
4	0	2	5	40	60	20	30	150

About Computer Graphics:

Computer graphics is a sub-field of computer science which studies methods for digitally synthesizing and manipulating visual content. Although the term often refers to the study of three dimensional computer graphics, it also encompasses two-dimensional graphics and image processing.

Basically there exists two kinds of computer graphics - raster (composed of pixels) and vector (composed of paths). Raster images are more commonly called bitmap images. A bitmap image uses a grid of individual pixels where each pixel can be a different color or shade.

The importance of computer graphics lies in its applications. Interactive computer graphics allows the physician to interpret this large volume of data in new and useful ways. Computer graphics has also expanded the boundaries of art and entertainment.

Course Overview:

In this course the students will learn about the concepts of computer graphics. It starts with an overview of interactive computer graphics, two dimensional system and mapping, then it presents the most important drawing algorithm, two-dimensional transformation; Clipping, filling and an introduction to 3-dimensional graphics.

Prerequisite:

This course does not require any programming background. This course helps the students to learn programming in python.

Course Outcome :

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

CO's	Course Outcome	Bloom Taxonomy

		Level
CO1	Understand Random and Raster scan systems, Graphics software and standards	Remember
CO2	Demonstrate Points, lines, circles and ellipses as graphics primitives	Understanding
CO3	Illustrate Fill area primitives including scan-line polygon filling	Understanding
CO4	Ability to draw basic 2-D geometric shapes	Applying
CO5	Ability to apply various 2-D transformations on geometric shapes	Applying
CO6	Ability to draw simple 3-D geometric shapes	Applying
CO7	Understand 3-D transformations and projections	Understanding
CO8	Demonstrate Points, lines, and polygon clippings	Applying

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PO13	PO14	PO15	PO16
CO1	1	3		3	3	3	2	2		1	1	1	2	3	1	3
CO2	2	3	3	3	1	1	2	2	1				2	3	1	3
CO3	3	3	3	3	3	3	2	3	2	2	2	1	2	3	2	2
CO4	1	3	3	3	3	3	3	2	2		1			3	1	2
CO5	1	3	3	3	3	2	1	2	1			1	2	2		2
CO6	2	1	1	3	3	1	1	2			2			2		1
CO7	1	1	2	1	2	1	2	1	1			1		2	1	
CO8	2		1	3	1	3	1	1	2		1		2		2	

Course Outline :

Unit #	Topics	Hours L+P
1	**Basic of Computer Graphics**: Basic of Computer Graphics: Coordinate Systems, Graphics APIs; Display devices, Random and Raster scan systems, Graphics software and standards; Color models: properties of light, XYZ, RGB, YIQ and CMY color models	8 + 2
2	**Graphics Primitives**: Points, lines, circles and ellipses as primitives, scan conversion algorithms for primitives, Fill area primitives including scan-line polygon filling, inside-outside test, boundary and flood-fill	10+10

3	**2D transformation and viewing**: Transformations (translation, rotation, scaling), matrix representation, homogeneous coordinates, composite transformations, reflection and shearing, viewing pipeline and coordinates system, window-to-viewport transformation, clipping including point clipping, line clipping (cohen-sutherland, liang- bersky, NLN), polygon clipping	10+10
4	**3D concepts and object representation**: 3D display methods, polygon surfaces, meshes, curved lies and surfaces, quadric surfaces, spline representation, cubic spline interpolation methods	10 + 8
5	**3D transformation and viewing**: 3D scaling, rotation and translation, composite transformation, parallel and perspective transformation, projection transformations	7 + 0
	Total	**45+30**

Reference :

Text Book :

1. Donald D. Hearn, M. Paulin Baker, & Warren Carithers, “Computer Graphics, with OpenGL”, Pearson Education
2. Sinha, Udai, “Computer Graphics”, Tata McGraw-Hill

Reference Books:

1. Foley, van Dam, “Computer Graphics”, Pearson Education
2. Francis S Hill, Jr. and Stephen M Kelley, “Computer Graphics Using OpenGL”, Prentice Hall
3. Peter Shirley, “Fundamentals of Computer Graphics”, A K Peters, 2009

Learning Resources:

SR.No	TEDx Video
T1	https://www.ted.com/talks/danielle_feinberg_the_magic_ingredient_that_brings_pixar_movies_to_life **Danielle Feinberg**, **Pixar's director of photography:** He creates stories with soul and wonder using math, science and code. Go behind the scenes of Finding Nemo, Toy Story, Brave, WALL-E and more, and discover how Pixar interweaves art and science to create fantastic worlds where the things you imagine can become real. This talk comes from the PBS special "TED Talks: Science & Wonder."
T2	https://www.youtube.com/watch?v=6hfOvs8pY1k **Animation basics: The art of timing and spacing - TED-Ed - David J. Malan** Expert timing and spacing is what separates a slide show from a truly amazing animation. TED-Ed demonstrates, by manipulating various bouncing balls, how the smallest adjustments from frame to frame can make all the difference. Lesson and animation by TED-Ed.

Other Videos:

Sr. No	**About Video**	**Link**	**Topic**

01.	Ken Joy is a Professor in thecomputer science department at the University of California at Davis.He has worked a number of years in the computer industry, and consults regularly on visualization, massive data analysis and geometric modeling.	https://www.youtube.com /w atch?v=01YSK5gIEYQ& list =PL_w_qWAQZtAZhtzPI 5p kAtcUVgmzdAP8g&inde x= 1	Preliminary background into some of the math associated with computer graphics.
02.	This complimentary course, originally presented at the SIGGRAPH 2013 conference	https://www.youtube.com /w atch?v=6-9XFm7XAT8	Introduction to programming OpenGL, emphasizing the most modern methods for using the library
03.	David E. Breen is currently an Associate Professor of Computer Science in the College of Computing and Informatics of Drexel University.	https://www.youtube.com /w atch?v=RDUH2412ZU0	In this talk I will introduce level set models and describe four computer graphics applications that utilize them. The applications are 3D morphing, contour-based surface reconstruction, volume segmentation and geometric modeling.
04.	Texas State Technical University	https://www.youtube.com /w atch?v=zG6j0be2E-0	The Graphics, Gaming & Simulation specialization of Computer Science Technology is designed to prepare students for entry into the world of graphics programming.
05.	James Abell , an artist and designer based in Scotland. He mix art, and traditional sculptures with digital technique.	https://www.youtube.com /w atch?v=seQuqguSiko	This video talks about the building blocks of 3d graphics. I talk about how they are worked out using the X/Y/Z axis. I then talk about primitives and then sub object modelling using the vertices/edges and polygons.

Related MOOCs courses

SR.No	MOOC Courses
M1	https://nptel.ac.in/courses/106/102/106102063/

M2	https://nptel.ac.in/courses/106/102/106102065/
M3	https://nptel.ac.in/courses/106/103/106103224/
M4	https://www.edx.org/course/foundations-computer-graphics-uc berkeleyx-cs-184-1x
M5	https://www.edx.org/course/computer-graphics-2
M6	https://www.coursera.org/learn/interactive-computer-graphics
M7	https://www.coursera.org/learn/geometric-algorithms
M8	https://www.udemy.com/course/graphics-with-modern-opengl/
M9	https://www.udemy.com/course/graphics-in-c/
M10	https://onlinecourses.nptel.ac.in/noc20_cs90/preview

Lab Experiments:

SR.NO	**Program Statement**
P1	Draw a line using Digital Differential Analyzer Line Algorithm [DDA]
P2	Draw a line using Bresenham's line drawing algorithm
P3	Draw a Circle using Bresenham's circle drawing algorithm
P4	Draw a Circle using Mid Point circle drawing algorithm
P5	Demonstrate Boundary Fill Algorithm
P6	Demonstrate Flood Fill Algorithm
P7	Demonstrate Two Dimensional Transformations
P8	Demonstrate Two Dimensional Composite Transformations
P9	Illustrate Cohen Sutherland Algorithm for a line clipping
P10	Illustrate Sutherland Hodgman Algorithm for a polygon clipping

Case Studies:

SR.No	**Titles**	**Evaluation Parameters for All**
C1	Google Sketch up	Identification of software Analysis and working methodology Output and result accuracy evaluation Report preparation Presentations
C2	FRASCA flight simulator	
C3	INTERSECTSimulator	

C4	Open Sees	
C5	HEC-RAS	

ITM (SLS) Baroda University
School of Computer Science, Engineering and Technology
B.Tech – Semester II

Course Name: Programming in Python-II
Course Code: C2210C4
Course Type: Core

Teaching Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
L	T	P	C	Theory Marks		Practical Marks		
				External	Internal	External	Internal	
2	0	4	4	40	60	20	30	150

What is OOP?

Object Oriented programming (OOP) is a programming paradigm that relies on the concept of classes and objects. It is used to structure a software program into simple, reusable pieces of code blueprints (usually called classes) which are used to create individual instances of objects.
A programmer designs a software program by organizing related pieces of information and behaviours together into a template called a class. Then individual objects are created from the class template. The entire software program runs by having multiple objects interact with objects to create the larger program.
OOP also prevents unwanted access to data, or exposing proprietary code through encapsulation and abstraction.

Course overview

In this course, students will learn how to do object oriented programming in Python. To handle exception and multithreading; Pattern matching with regular expressions, to handle system commands and testing in python, understand the basics of data science libraries numpy,pandas and matplotlib.

Outcome Applying

Problem t Oriented

Statemen Object Concept

Necessary Libraries	and Modules	Program Developm	e nt

Prerequisite

Programming in Python-I

Learning outcomes:

After completing the course, the student shall be able to :

	Course Outcome	**Bloom's Level**
CO1	Understand basics concepts of object oriented programming using python	Understanding
CO2	Learn to handle exceptions and errors in python	Application
CO3	Learn the execution in multithreaded environment	Applying
CO4	Understand the regular expression and its applications	Creation
CO5	Dealing with basic data science libraries numpy, pandas and matplotlib	Creation
CO6	Understand the system level interaction and testing with python	Applying

CO-PO Mapping

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO10**	**PO11**	**PO12**	**PO13**	**PO14**	**PO15**	**PO 16**
CO1	**3**	**3**	**3**	**2**	**3**	**3**	**2**	**2**	**2**	__	__	__	**1**	**3**	**2**	**2**
CO2	**2**	**3**	**3**	**3**	**3**	**3**	**2**	**2**	**1**	__	__	__	__	**3**	**1**	**2**
CO3	**3**	**3**	**3**	**2**	**3**	**3**	**3**	**2**	**1**	__	__	__	__	**3**	**1**	**3**
CO4	**3**	**3**	**3**	**2**	**1**	**2**	**1**	**1**	__	__	__	__	__	**2**	**1**	**1**
CO5	**3**	**3**	**3**	**3**	**2**	**3**	**3**	**3**	**2**	_	**1**	**1**	**1**	**2**	**1**	**3**
CO6	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**1**	**1**	__	__	**1**	**3**	**2**	**3**

Course Outline:

U ni t #	Topics		Lab As	Hours
1	**OOP :** Introduction of Object-oriented Basic Concepts: Class, Objects, Constructor, Destructor. Methods for updating set...() and retrieving get...() of data members of the class, special methods, Use of self method, _init_ method. **Introduction to Polymorphism and Encapsulation**: Operator Overloading, and Function Overloading, Introduction to Encapsulation and Data Hiding. **Inheritance and Abstraction :** Modifier and Accessor Methods; Introduction to Inheritance; Single Inheritance; Hierarchical Inheritance; Multiple Inheritance; and Data Abstraction; Abstract Methods and Abstract class, Attribute Resolution Order for Inheritance; Built-in Functions for Classes	P1, P 2, P3 , P4	A1,A2	10
2	**Exception Handling:** Handling an exception, Exception Hierarchy, The Exception Model, Run Time Errors, try…….except…….else, try-finally-clause, Argument of an exception, Python standard exceptions and user defined exceptions, Handling IO Exceptions. **Multithreading:** Starting a new thread, the threading module, synchronizing threads, race condition, multithreaded priority queue.	P5, P 6	A3	8
3	**Regular Expressions**: Introduction/Motivation, Special Symbols and Characters for REs, REs and Python, Handling regular expressions and their built-in methods, Lambda Expression, Map(), Reduce(), Filter().	P7	--	8
4	**Programming with Files:** Reading files,iterating through filesfiles,working with files, more file information, directories,reading CSV Files,GeneratingCSV,Reading and Writing CSV Files with Dictionaries	P8	--	2

5	**Overview of Numpy:**Introduction of Array, List vs. NumPy, applications of NumPy, How to Install NumPy, working with shape,reshape and dimension using NumPy, vertical and horizontal stacking, ones and zeros array,Indexing and Slicing NumPy Arrays in Python, other in-built function of NumPy	P8	A4	6

6	**Overview of Pandas and Matplotlib:**Introduction to Pandas,How to install pandas,Applications of Pandas,Dataframes and Series, reading csv, Excel and text files using pandas, getting description of dataset and simple functions of pandas. Introduction to Matplotlib with its application, using label, color and markers, axis, scatter plot, bar graph, histogram and line plot.	P9, P10	A5,A6	6
7	**Interacting at System Level:** Reading Data interactively,Standard streams, environment variables, command line arguments and exit status, running system commands in python, obtaining the output of a system command.	—	—	2
8	**Testing in Python:** What is testing?, Manual Testing and Automated Testing, Unit test, Writing Unit Test in Python, Edge Cases, Additional test cases, Black Box vs. White Box, other test types, test driven development	P11	__	4
	TOTAL			**46**

Reference:

Textbook:

Sr.No	**Book Name**
1	SheetalTaneja, Naveen Kumar, “Python Programming: A Modular Approach”, Pearson (2019)
2	R. Nageswara Rao, “Core Python Programming”, dreamtech

Reference Book:

Sr.No	**Book Name**

1	ReemaThareja, “Python Programming: Using Problem Solving Approach”, Oxford University Press (2017)
2	John V Guttag, “Introduction to Computation and Programming using Python with Application to Understanding Data”, PHI (2016)
3	Martin C Brown, “Python: The Complete Reference”, McGraw-Hill Education (2018)
4	YashavantKanetkar, “Let us Python”, BPB Publication (2019)
5	Stephen Klosterman, “Data Science Projects with Python: A case study approach to successful data science projects using Python, pandas, and scikit”, Packt Publishing (2019) Kindle edition
6.	Python Data Science Handbook: Essential Tools for Working with Data Paperback 2016by Jake VanderPlas (Author)
7.	Python for Everybody: Exploring Data in Python 2016by Dr. Charles Russell Severance (Author), Sue Blumenberg (Editor), Elliott Hauser (Editor)

Required Software:

1. Python Version 3.6https://www.python.org/downloads/
2. Google Colab: https://colab.research.google.com/notebooks/io.ipynb

Learning Resources:

Learning Related Website: https://www.python-course.eu/

TedEx Videos:

Sr. No	TEDx Video
T1	World Changing: Data Science and AI | Fred Blackburn | TEDxURL The lecture covered the key points like tremendous increase in data, real world examples of machine learning i.e Alexa, Robot Scientist, Healthcare industry, artwork and many more.Video also covered the racing trends of living and working with human intelligence and the learning pattern of human mind and machine
T2	The most important skills of data scientists | Jose Miguel Cansado | TEDxURL In this talk Jose Miguel Cansado had discussed how the data should be extracted to give the meaningful insights and then gives the analogy of human factors in data mining to become a good data scientist as well as the applications of machine learning and AI

Other Videos:

Sr. No	About Video	Link	Topic
01	Dr. Anna Bell,MITopencourseware	https://www.youtube.com/watch?v=-DP1i2ZU9gk	Object Oriented Programming
02	Prof. Eric Grimson, Prof. John Guttag, MIT Open Courseware	https://www.youtube.com/watch?v=y81AhLQNNI&list=PL38986218EEA17EB3&index=15	Abstract Data Types, Classes and methods
03	Plotting in Python by Prof. Eric Grimson, MIT	https://www.youtube.com/watch?v=D3-dPiswfEY&list=PLRJdqdXieSHN0U9AdnmwD9QcR9hmw04d&index=78	Matplotlib
04	Prof. NipunBatra,IIT Gandhinagar	https://www.youtube.com/watch?v=e2zCQjiFXgE	NumPy

Related MOOCs courses:

Sr.No	MOOC Courses
M1	“The Joy of Computing Using Python” by Prof. SudarshanIyengar, IIT Ropar 12 Weeks on NPTEL.
M2	“Python for Data Science” by Prof. RagunathanRengasamy, IIT Madras 4 weeks on NPTEL.

Lab Experiments:

Sr. No.	Program Statement

P1	Write Python programs to do the following: 1. Create a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle. 2. Create a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle. Take different instances (objects) of class Rectangle and apply the above methods. Also, measure the time and space complexity of the program.
P2	1. Write a Python program to implement operator overloading in python.Also, measure the time and space complexity of the program. 2. Write a Python program to implement function overloading in python.Also, measure the time and space complexity of the program.
P3 2.	**Inheritance:** 1. Write a Python program to perform single level and multiple inheritance in python. Derive class Student from class Person, define additional attributes: enrollmentNo, discipline, gotMarks, result. Derive class Graduate and class Postgraduate from class Student. Define methods: percentage() and result() in class Student, class Graduate, and class Postgraduate. Class Student will use default values for maxMarks = 500, and passPercentage = 35%; for class Graduate, maxMarks = 600, and passPercentage = 40%; for class Postgraduate, maxMarks = 400, and passPercentage = 50%.Also, measure the time and space complexity of the program.
P4	**Abstraction:** Define class Shape by inheriting class Circle, class Rectangle, and class Triangle.
	Redefine methods: perimeter() and area() for class Shape using the concept of abstraction.Measure the time and space complexity of the program.
P5	**Exception Handling:** 1. Write a Python program that accepts two numbers from the user and performs their division and if the divisor is zero then raise ZeroDivisionError. The code should demonstrates the uses of else and finally blocks as well. 2. Write a Python programthat accepts a number from the user. The try block should raise a ValueError exception if the number is outside the allowed range. 3. Write a Python program to raise an exception of IndexError python.

P6	**Multithreading:** 1. Write a Python program to demonstrate the concept of multithreading in python. 2. Write a Python program to create two functions which will calculate the cube and square of any given number and should apply the concept of multithreading with join, start and thread information of both the functions.
P7	**Regular Expression:** 1. Write a Python program to check whether a specified character is at the start or end of the word for a given string using regular expression module in python. 2. Write a Python program to find all the numbers present in a given string entered by user using regular expression. Input: “Hello 100 Hii 500, How is your day 134” Output: [‘100’,’500’,’34’]
P8	**NumPy:** 1. Create a 4X2 integer array and Prints its attributes and print the following attributes: – • The shape of an array. • Array dimensions. • The Length of each element of the array in bytes. 2. Take two matrices and perform addition, subtraction and multiplication of two matrices. 3. Following is the 2-D array. Print max from axis 0 and min from axis 1 ([[34,43,73],[82,22,12],[53,94,66]]) **Expected Output:** for min of axis 1 [34 12 53] for max of axis 0 [82 94 73]
P9	**Pandas:** 1. Write Python program to create series and dataframe using pandas. 2. Display the top 10 and bottom 7 values from your dataframe. 3. Load data from a comma separated values(CSV) file into pandas and display its information and the unique values present in each column of the dataset.

P10	**Matplotlib:** Draw bar, histogram and scatter plot using matplotlib.
P11	**Unit Testing:** Write a Python program to perform unit testing on a module cuboid volume created by user for any negative number if it is eneterd by user.

Assignments:

Sr. No.	**Assignment Name**
A1	Create a class called apartment with attributes flatnumber,ownername,electicity bill amount. Create another class apartment_demo with definit(self): pass to create a method getSecondMinBill that takes the list of objects and gives the second minimum electricity bill as output. **Input:** 3(no of objects to be created) 1000 Hari 5000 1001 Hena 5002 1002 Harsha 5001 **Output:** 5001 (since it is the second minimum bill amount among the bills)
A2	Create a class Bill with attributes mobile number and payment bill. Create another class mobile with attributes service provider, mobile number, data used, payment method. Service provider maybe airtel or jio. Data used is integer values in Gigabytes(GB). Payment method maybe paytm,gpay,amazon and so on. Create a method calculate bill that takes the list of objects and calculates the bill and returns the list of objects of class bill with mobile number and payment bill.

	The payment is calculated as follows: 1.If the service provider is airtel, the bill is Rs.11 for every 1GB used and if it is jio, the bill is Rs.10 for every 1GB used. 2. If the payment method is paytm there is a cashback of 10% of the total bill for airtel users only. The bill is calculated and rounded off after deducing the cashback value. **Input:** 3(No of objects to be created) airtel 123 16 paytm airtel 456 10 amazon jio 788 10 paytm **Output:** (123,158) (456,110) (789,100)
A3	Create a user defined exception for a salary enteredbyuser should be in range between 5000 and 15000.
A4	How to get the positions where elements of two arrays match.
A5	Create a bar graph to show the average attendance of a class division wise.
A6	Load IMDB dataset using pandas and display all the necessary information related to the dataframe.

ITM(SLS) Baroda University
School of Computer Science, Engineering and Technology
B.Tech - Semester III

Course Name :Object Oriented Programming with Java
Course Code:C2310C1
Course Type: Core

Teaching Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
L	**T**	**P**	**C**	**Theory Marks**		**Practical marks**		
				External	**Internal**	**External**	**Internal**	
3	**0**	**4**	**5**	**40**	**60**	**0**	**50**	**150**

Preamble-The purpose of this course is to enable learners to solve problems by breaking it down to object level while designing software and to implement it using Java. This course covers Object Oriented Principles, Object Oriented Programming in Java, Inheritance, Exception handling, Event handling, multithreaded programming and working with window-based graphics. This course helps to develop Desktop GUI Applications, Mobile applications, Enterprise Applications, Scientific Applications and Web based Applications.

Prerequisite: Topics covered under the course Programming in C,Programming in Python-1 , Programming in Python-2.

What is Object Oriented Programming - Object-Oriented Programming is a methodology or paradigm to design a program using classes and objects. It simplifies software development and maintenance by providing some concepts:

- Object
- Class
- Inheritance
- Polymorphism
- Abstraction
- Encapsulation

Course Objective:

1. Understand fundamentals of JAVA programming such as variables, conditional and iterativeexecution, methods, etc.
2. Understand fundamentals of object-oriented programming in Java, including definingclasses, invoking methods, using class libraries, etc.

3. Be aware of the important topics and principles of software development.
4. Have the ability to write a computer program to solve specified problems.
5. Be able to use the Java SDK environment to create, debug and run simple Java programs. 6. Understand the concept of JAVA FX basics Control.

Course Learning Outcome:

After completing the course, the student shall be able to :

	Course Outcome	Bloom's Level
CO1	Use various Java constructs, features and libraries for simple problems.	Understanding
CO2	Demonstrate how to define and use classes, interfaces, create objects and methods, how to override and overload methods.	Understanding
CO3	Comprehend building blocks of OOPs language, inheritance, package and interfaces	Applying
CO4	Identify exception handling methods& JAVA FX Controls	Application
CO5	Implement multithreading in object oriented programs.	Application

Course Competency:

1. Understanding of the Java system architecture and its major components by Java Runtime
2. Environment (JRE) and the Java Development Kit (JDK).
3. Understanding of the object oriented programming concepts.
4. Understanding of the multithreading in object oriented programs.
5. Understanding of the exception handling methods & JAVA FX Controls.

Course Contents:

Unit #	*Topics (Programs) to be Completed*	*Total Hrs.*
1	**Basics of Java**: Features of Java, Byte Code and Java Virtual Machine, JDK, Data types, Operator, Control Statements –If , else, nested if, if-else ladders, Switch, while, do-while, for, for-each, break, continue. **Array and String**: Single and Multidimensional Array, String class, StringBuffer class, Operations on string, Command line argument, Use of Wrapper Class.	8
2	**Classes, Objects and Methods**: Class, Object, Object reference, Constructor, Constructor Overloading, Method Overloading, Recursion, Passing and Returning object form Method, new operator, this and static keyword, finalize() method, Access control, modifiers, Nested class, Inner class, Anonymous inner class.	8
3	**Inheritance and Interfaces:** Use of Inheritance, Inheriting Data members and Methods, constructor in inheritance, Multilevel Inheritance –method overriding Handle multilevel constructors –super keyword,Stop Inheritance -Final keywords, Creation and Implementation of an interface, Interface reference, instanceof operator, Interface inheritance, Dynamic method dispatch , Comparison between Abstract Class and interface.	6

4	**JAVAFX UI controls and multimedia:** Labeled and Label, button, Checkbox, Radio Button, Text field, Text Area, Combo Box, List View, Scrollbar, Slider, Video and Audio. **Exception Handling**: Exception and Error, Use of try, catch, throw, throws and finally, Built in Exception, Custom exception, Throwable Class.	6
5	**Multithreaded Programming:** Use of Multithread programming, Thread class and Runable interface,Thread priority, Thread synchronization and Locks, Thread communication, Deadlock.DeamonThread.Methods of thread . **IO Programming**: Introduction to Stream, Byte Stream, Character stream, Readers and Writers, File Class, File Input Stream, File Output Stream, Input StreamReader, OutputStreamWriter, FileReader, FileWriter, Buffered Reader **Class Modeling**: Object, class concepts, link and association, Generalization and Inheritance.	8
		40

Reference :

Text Books:

1. Core Java Volume-I Fundamentals Horstmann& Cornell, -Pearson Education. -Eight Edition

2. Object Oriented Modelling and Design with UML Michael Blaha and James Rambaugh –PEARSON Second edition

Reference Books:

1. Java Fundamentals A comprehensive introduction By Herbert Schildt, Dale Skrien, McGraw Hill Education.
2. Programming with Java A Primer –E.Balaguruswamy,McGrawhill.
3. The Complete Reference, Java 2 (Fourth Edition),HerbertSchild, -TMH.
4. The Class of Java-Mr Praveen Jain

Case Studies:

Sr.No	**Case Studies**	**Evaluation**
C1	Development of Content Management System	1. Identification of algorithm. 2. Report preparation. 3. Presentation with VIVA

C2	Development of Employee Performance Software System	

Simulation and Animation: NA

TEDx Videos:

Sr. No	TEDx Video
T1	1. https://www.youtube.com/watch?v=qGW0GT1rCvs - A programming language to heal the planet together: Julia	Alan Edelman	TEDxMIT

NPTEL Video:

Sr. No	About Video	Link	Topic
O1	Concept of Java Programming Language Dr.DebasisSanmant,IITKhara gpur	https://www.youtube.com/watch?v=VksxhzfD8kQ&list=PLfn3cNtmZdPOe3R_wO _h540QNfMkCQ0ho&index=2	Core Concept
O2	Java Tools and Resourses	https://www.youtube.com/watch?v=1B5ppTif5ZY&list=PLfn3cNtmZdPOe3R_wO _h540QNfMkCQ0ho&index=3	Jdk Tools
O3	Java Applet Programming	https://www.youtube.com/watch?v=0pzR2FGTEhk&list=PLfn3cNtmZdPOe3R_w O_h540QNfMkCQ0ho&index=5	Applet Programming

Other Videos:

Sr. No	About Video	Link
O1	Java Framework	https://www.youtube.com/watch?v=cRL5TorHINY&list=PL7 WFbgpeASD3117pvlNaXZwK5q6H3XhZZ
O2	Java Installation Guide.	https://www.youtube.com/watch?v=eixYN5v7jOY&list=PL7 WFbgpeASD3117pvlNaXZwK5q6H3XhZZ&index=2

O3	Java Programming Steps	https://www.youtube.com/watch?v=VksxhzfD8kQ&list=PLfn 3cNtmZdPOe3R_wO_h540QNfMkCQ0ho&index=2

Related MOOCs courses

Sr.No	MOOC Courses
M1	Programming, Java Programming by Prof.DebasisSamant 8 Weeks on NPTE
M2	Java Programming Specialization Offered By University of California San Diego National Research University Higher School of Economics(coursera)

Activity Based Learning(ABL):

S.NO	Name of Activity	Details of Activity	Outcome	Evaluation
01	**Chat Application Development**	A chat application is the process of exchanging messages between two systems continuously.It is a console application that is launched from the command line. The server and clients can run on different computers in the same network, e.g. Local Area Network (LAN).This application uses the concepts of Swing GUI widget toolkit, multithreading and Socket programming for communication over network.	Upon completion of this ABL, the students will be able to: • Perform basic GUI designing • Multitasking application development can be done using concept of multithreading • Communication over the network can be learned	Based on Chat application development process

Lab Experiments:

S.No	Experiment Name	Total Hrs
PS1	Write a Program that displays Welcome to Java, Learning Java Now and Programming is fun.	1
PS2	Write a program that solves the following equation and displays the value x and y: 1) 3.4x+50.2y=44.5 2) 2.1x+.55y=5.9 (Assume Cramer's rule to solve equation ax+by=e x=ed-bf/ad-bc cx+dy=f y=af-ec/ad-bc)	1

PS3	Write a program that reads a number in meters, converts it to feet, and displays the result.	1/2
PS4	Body Mass Index (BMI) is a measure of health on weight. It can be calculated by taking your weight in kilograms and dividing by the square of your height in meters. Write a program that prompts the user to enter a weight in pounds and height in inches and displays the BMI. Note:- 1 pound=.45359237 Kg and 1 inch=.0254 meters.	1
PS5	Write a program that prompts the user to enter three integers and display the integers in decreasing order.	1
PS6	Write a program that prompts the user to enter a letter and check whether a letter is a vowel or constant.	1
PS7	Assume a vehicle plate number consists of three uppercase letters followed by four digits. Write a program to generate a plate number.	1
PS8	Write a program that reads an integer and displays all its smallest factors in increasing order. For example if input number is 120, the output should be as follows:2,2,2,3,5.	1
PS9	Write a method with following method header. public static intgcd(int num1, int num2) Write a program that prompts the user to enter two integers and compute the gcd of two integers.	1
PS10	Write a test program that prompts the user to enter ten numbers, invoke a method to reverse the numbers, display the numbers.	1
PS11	Write a program that generate 6*6 two-dimensional matrix, filled with 0's and 1's , display the matrix, check every raw and column have an odd number's of 1's.	2
PS12	Write a program that creates a Random object with seed 1000 and displays the first 100 random integers between 1 and 49 using the NextInt (49) method.	2
PS13	Write a program to implement Inheritance & its types	2
PS14	Write a program to implement Exception handling	2
PS15	Write a Program to implement multithreaded program	2
PS16	Write a GUI program that use button to move the message to the left and right and use the radio button to change the colour for the message displayed	2
PS17	Write a Program to implement Class Modeling	2

Mini Projects:

Sno.	**Title**

MP1	A Semantic Web-Based Scientific News Aggregator
MP2	Develop a Product on Advance Courier Service
MP3	Creation of Advance Web-based Multimedia Answer Generation

ITM(SLS) Baroda University
School of Computer Science, Engineering and Technology
B.Tech- Semester III

Course Name: Computer Architecture
Course Code: C2310C2
Course Type: Core

Teaching Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
L	T	P		Theory		Practical		
				External	Internal	External	Internal	
3	0	2	4	40	60	0	50	150

Course Introduction

This course provides detail of computer system's functional components, their characteristics, performance and interactions including system bus, different types of memory and CPU. This course also covers the architectural issues such as instruction set program and data types. The course emphasizes performance and cost analysis, instruction set design, pipelining, memory technology, memory hierarchy, virtual memory management, and I/O systems

Course Objective:

1. To understand the structure, function and characteristics of computer systems. 2. To understand the design of the various functional units and components of digital computers. To identify the elements of modern instructions sets and explain their impact on processor design.
3. To explain the function of each element of a memory hierarchy, identify and compare different methods for computer I/O.
4. To compare simple computer architectures and organizations based on established performance metrics.

Competencies:

The course content should be taught and implemented with the aim to develop different types of skills so that students are able to acquire following competencies:

1. Apply computer architecture theory to solve the basic functional computer problem.
2. Show and assemble basic computer components.
3. Ability to integrate into working groups involved in analysis and design tasks

Course Learning Outcome:

After completing the course, the student shall be able to :

	Course Outcome	Bloom's Level

CO1	Identify and explain the basic structure and functional units of a digital computer	Understanding
CO2	Identify the role and working of various functional units of a computer for execution of instruction	Understanding
CO3	Design processing unit using the concepts of ALU and control logic design.	Design
CO4	Design interfacing of memory and I/O modules with CPU	Design
CO5	Implement assembly language programs and execute them	Implement
CO6	Compare performance of different types of computer architectures	Analyze

Course Syllabus:

Unit #	**Topic Name**	**No of Hours**
1	**Data storage and register transfer operations** Register Transfer and Micro-operations: Register Transfer language, Register Transfer, Bus and Memory Transfers, ArithmeticMicrooperations, Logic Micro-Operations, Shift Micro-Operations, Arithmetic logical shift unit	5
2	**Basic Computer Organization and Design** Instruction codes, Computer registers, computer instructions, Timing and Control, Instruction cycle, Memory-Reference Instructions, Input-output and interrupt, Design of Basic computer, Design of Accumulator Unit.	6
3	**Assembly Language Programming** Introduction, Machine Language, Assembly Language Programming: Arithmetic andlogic operations, looping constructs, Subroutines, I-O Programming.	7
4	**Microprogrammed Control Organization:** Control Memory, Address sequencing, Micro program example, Design of Control Unit	3

5	**Central Processing Unit** Introduction, General Register Organization, Stack Organization, Instruction format,Addressing Modes, Data transfer and manipulation, Program control, ReducedInstruction Set Computer (RISC) & Complex Instruction Set Computer (CISC)	5
6	**Pipeline And Vector Processing**Flynn's taxonomy, Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processors,	5
7	**Computer Arithmetic** Introduction, Addition and subtraction, Multiplication Algorithms (Booth MultiplicationAlgorithm), Division Algorithms, Floating Point Arithmetic operations,	4
8	**Input-Output Organization** Input-Output Interface, Asynchronous Data Transfer, Modes of Transfer, PriorityInterrupt, DMA, Input-Output Processor (IOP), CPU IOP Communication, Serialcommunication.	5
9	**Memory Organization** Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Virtual Memory, Introduction to GPU.	5
	Total Hours	**45**

References:

Textbook: -

1. M. Morris Mano, “Computer System Architecture”, Pearson Education

References:

1. Yale N. Patt, Sanjay J. Patel, “Introduction to Computing Systems” McGraw Hill
2. Hamacher, Vranesic, Zaky, “Computer Organization”, McGraw Hill.
3. Andrew S. Tanenbaum and Todd Austin, “Structured Computer Organization”, Pearson Education
4. N. D. Jotwani, “Computer system organization”, McGraw Hill
5. R.S.Gaonkar, “Microprocessor Architecture, Programming and Applications with 8085A”, Penram International
6. Douglas Hall, Microprocessors and Interfacing, TMH

TEDx Videos:

Sr No	**Link details**	**Description**

TD1	https://www.ted.com/talks/kanawat_se nanan_how_computer_memory_wo rk s?language=en	How computer memory works. Kanawat SenananIn many ways, our memories make us who we are, helping us remember our past, learn and retain skills, and plan for the future. And for the computers that often act as extensions of ourselves, memory plays much the same role.
TD2	https://www.ted.com/talks/bettina_b air _inside_your_computerBettina Bair	How does a computer work? The critical components of a computer are the peripherals (including the mouse), the input/output subsystem (which controls what and how much information comes in and out), and the central processing unit (the brains), as well **as human** written programs and memory

Other Videos:

Sr. No	About Video	Link	Topic
01.	New Golden Age for Computer Architecture: Domain-Specific Hardware/Software Co-Design, Enhanced Security, Open Instruction Sets, and Agile Chip Development Speaker: John Hennessy, 2017 Turing Award Recipient / Chairman, Alphabet	https://www.youtube.com/watch?v=bfPV4x -HrUI	New Golden Age for Computer Architecture
02.	This video explains how the pipelining concept is used to implement various tasks.	https://www.youtube.com/watch?v=3p8kZp T56lQ	Pipelining introduction
03	Digital Circuits & Systems by Prof. S. Srinivasan, Department of Electrical Engineering, IIT Madras This video explains how the arithmetic circuit will perform various operations.	https://www.youtube.com/watch?v=NAqR OGjgoQ	Arithmetic circuit

04	Assembly Language & Computer Architecture Prof. Leiserson Stages of code from source code to compilation to machine code to hardware interpretation and, finally, to execution.	https://ocw.mit.edu/courses/electrical engineering-and-computer-science/6-172- performance-engineering-of-software systems-fall-2018/lecture-videos/lecture-4- assembly-language-computer-architecture/	Assembly Language & Computer Architecture
05	Storage and I/O Interface Prof. Jatindra Kumar Deka Department of Computer Science and Engineering, IIT Guwahati. This video describes the communication between input and output devices and storage devices	https://www.youtube.com/watch?v=cipkW LPAsKE	Storage and I/O Interface
06	Cache memory, also called CPU memory, is random access memory (RAM) that a computer microprocessor can access more quickly than it can access regular RAM. This memory is typically integrated directly with the CPU chip or placed on a separate chip that has a separate bus interconnect with the CPU	https://www.youtube.com/watch?v=46dfG0 nW3v4	CACHE MEMORY ANIMATION : Computer Architecture Concepts

Related MOOCs courses

M1	Computer Organization and Architecture: A Pedagogical AspectDr. Arnab Sarkar IIT Gowahati by NPTEL https://onlinecourses.nptel.ac.in/noc21_cs37/preview
M2	Computer Organization and Architecture by VmKamkoti by IIT Madras by NPTEL. https://nptel.ac.in/courses/106/106/106106166/
M3	Computer Architecture by David Wentzlaff. By Coursera https://www.coursera.org/learn/comparch?action=enroll

Additional Resources: - (Case Studies)

Sr No	Case Studies	Evaluation

C1 C2 C3	Recent Intel processor Pipelining in Pentium RISC-V	1 Report preparation. 2. Presentation with VIVA

Lab Experiments:

Sr No	Title of Experiments
P1	Construct the logical Half and full adder
P2	Construct the Logical Diagram for tri state bus buffers circuit using Logisim
P3	Design the computational circuit for status register
P4	Design computational circuit for Basic traffic Signal
P5	Design a 16 to 1 line multiplexer with 8-to-1-line multiplexers and one 2 to 1 line multiplexers.
P6	Design the computational circuit for 4bit arithmetic circuit
P7	Design the computational circuit for binary to Hexadecimal Conversion using 7 Segment Display
P8	Construct the Logical Diagram to perform shift operations using Logisim
P9	Write an assembly language program for performing arithmetic operations
P10	Write an assembly language program to perform subroutine
P11	Write an assembly language and factorial of number
P12	Write an assembly language to sum of even numbers
P13	Write a program to calculate the sum of series of number

ITM(SLS)BarodaUniversity
SchoolofComputerScience,EngineeringandTechnology
B.Tech - Semester III

Course Name: System Software
Course Code: C2310C3
CourseType: Core

TeachingScheme :

TeachingScheme			Credits	ExaminationMarks				Total Marks
L	T	P	C	Theory Marks		Practicalmarks		
				External	Internal	External	Internal	
3	0	2	4	40	60	0	50	150

Preamble:

This course is concerned with those software tools and utilities which are an essential part of a Computer System which supports application software development. In order to apply a computer system to solve problems and provide computational services for a wide variety of human activities we need Application Software, e.g. an Accounting System package. However, to develop such an application package we need System Software like text Editors, Language processors - Compilers, Interpreters and Assemblers, several utilities like Linkers and program Loaders, and File Systems. Out of these the File System is generally studied as a part of Operating Systems, due to its close links with components of an Operating System and computer Hardware. The rest forms parts of the present course.

What is system Software?

Systems Programming is the activity of writing and maintaining system software. System software is the layer between the hardware and application software; it controls the hardware,and provides services to applications.
The primary distinguishing characteristic of systems programming when compared to application programming is that application programming aims to produce software which provides services to the user directly (e.g. word processor),
Systems programming is the design, implementation, and maintenance of all of the programs like Operating System, assembler, linker, kernel, Bios, Macro Processor, Loader, Assembler etc.….

Prerequisite: Basis of Data Structures and Basic concepts of Operating System
Course Learning Outcome:
After completing the course, the student shall be able to:

	Course Outcome	Bloom's Level

CO1	Ability to understand Software and Its types and use text editors	Understanding
CO2	Ability to understand the Overview of Language Processors	Understanding
CO3	Ability to Understand the concept of Compiler	Understanding
CO4	Ability to Understand the Assembler& Implement Linker and Loaders	Applying
CO5	Ability to use and Implement Macro & Macro Processor	Application
CO6	Ability to understand Utility of Interpreter and Debuggers	Application

Course Syllabus:

Tasks #	*Topics (Programs) to be Completed*	*Total Hrs.*
1	**Overview of System Software and Text Editors:** Introduction, Software, Software Hierarchy, Systems Programming Tools, Life Cycle of a Source Program, Levels of System Software Text Editors: Overview of Editing Process, User Interface, Editor Structure, Text Editors-line-by-line (example ed), file-oriented WYSIWYG (example vi)	6
2	**Overview of Language Processors**: Programming Languages and Language Processors, Language Processing Activities, Program Execution, Fundamental of Language Processing, Symbol Tables Data Structures for Language Processing: Search Data structures, Allocation Data Structures**.**	6
3	**Assemblers** Elements of Assembly Language Programming, Design of the Assembler, Assembler Design Criteria, Types of Assemblers, Two-Pass Assemblers, One-Pass Assemblers **Macro and Macro Processors** Introduction, Macro Definition and Call, Macro Expansion, Nested Macro Calls, Advanced Macro Facilities, Design of a Macro Pre-processor, Design of a Macro Assembler, Functions of a Macro Processor, Basic Tasks of a Macro Processor, Design Issues of Macro Processors	10
4	**Overview of the Compiler:** A Simple Compiler, Difference between interpreter, assembler and compiler. types of Compiler, Analysis of the Source Program, The Phases of a Compiler, parsing techniques, The Grouping of Phases-Front end and back end of compiler., Parsing techniques, Code optimization Techniques	10
5	**Interpreter & Debuggers:** Benefits of Interpretation, Overview of Interpretation, The Java Language Environment, Java Virtual Machine, Types of Errors, Debugging Procedures, Classification of Debuggers, Dynamic/Interactive Debugger.	4

6	**Linker & Loader:** Introduction, Relocation of Linking Concept, Design of a Linker, Self-Relocating Programs, linking in Linux, Linking of Overlay Structured Programs, Dynamic Linking, Loaders, Different Loading Schemes, Sequential and Direct Loaders, Compile-and-Go Loaders	8
	Total	**42**

References:

Text Books:

Sr.No Book Name

1 System Software –An Introduction to Systems Programming by Leland L. Beck, 3rd Edition, Pearson Education Asia, 2000

2 System Software by Santanu Chattopadhyay,Prentice-Hall India,2007

References:

Sr.No Book Name

1 “Compilers - Principles and Practice”, P.H.Dave and H.B.Dave, Pearson Education 2. “System Programming and Operating System”, 2nd Ed, D.M.Dhamdhere, Tata McGraw Hill.

3. System Programming by Srimanta Pal OXFORD Publication

4. System Programming and Compiler Construction by R.K. Maurya & A. Godbole

Required Software:

1. GNU C Compiler: https://gcc.gnu.org/
2. For working with embedded system corresponding architecture tool chain : Assembler,Link editor, etc.

Web Resources:

1. https://texteditors.org/cgi-bin/wiki.pl
2. https://www.geeksforgeeks.org/editors-types-system-programming/
3. https://www.ques10.com/p/8931/write-a-brief-note-on-design-of-an-editor-2/
4. https://www.programming1011.com/2019/04/short-note-on-editors-in-system.html
5. http://www.tezu.ernet.in/~utpal/course_mat/ss_editor.html

Case Studies:

Sr. No	**Case Studies**	**Evaluation**

C1	Various compiler tools like GNU tool chain	1. Identification of algorithm. 2. Report preparation. 3. Presentation with VIVA
C2	Lexical Analyzer and YACC tools	

Related MOOCs courses

Sr. No	Course
M1	Introduction to System Software by Prof.S. Raman, Department of Computer Science andEngineering, IIT Madras. For More details on NPTEL visit http://nptel.iitm.ac.in

Other Videos Links:

Sr. No	About Video	Link
O1	Grammar and Its types	https://www.coursera.org/lecture/nand2tetris2/unit-4-3-grammars rtIKX
O2	Syntax Analysis	https://www.coursera.org/lecture/nand2tetris2/unit-4-1-syntax analysis-5pC2Z
O3	Lexical Analysis	https://www.coursera.org/lecture/nand2tetris2/unit-4-2-lexical analysis-QM0lZ
04	Assembler construction by NPTEL-NOC IITM	https://www.youtube.com/watch?v=LEKDkhvUKjA
05	Front end loader by NPTEL IIT Guwahati	https://www.youtube.com/watch?v=DBTkILIZmSc

Lab Experiments:

S.No	Program Statement	Hrs
PS1	Write a C program to identify whether a given line is a comment or not	1
PS2	Write a C program to test whether a given identifier is valid or not.	1
PS3	Write a C program to simulate lexical analyzer for validating operators	1

PS4	Implement following programs using Lex. a. Program to count the number of characters, words, spaces and lines in a given input file. b. Program to count the numbers of comment lines in a given C program. Also eliminate them and copy the resulting program into separate file. **2) a.** Program to recognize a valid arithmetic expression and to recognize the identifiers and operators present. Print them separately. **b.** Program to recognize whether a given sentence is simple or compound. **3)** Program to recognize and count the number of identifiers in a given input file.	2
PS5	Use macro features of C language	2
PS6	Write a program to left factor the given grammar	2
PS7	Lex program to print out all numbers from the given file.	2
PS8	Write a program to remove the Left Recursion from a given grammar	2
PS9	Write a program which generates Quadruple Table for the given postfix String	2
PS10	Write a C program to parse a given string using Predictive parsing for given grammar. Type→ simple | ↑id | array [simple] of type simple → integer | char | num dot dot num	2
PS11	Write a program to remove the Left Recursion from a given grammar	2
PSI2	Write a Program to identify all the tokens from the source code	2
PS13	Write a Program for first pass and second pass Assembler	2
PS14	Write a program to show linker.	2
PS15	C program to generate in triple intermediate code for assignment statement.	2
	Total	**27**

Mini Projects:

Sr.no.	
MP1	C program to generate triple for assignment statement.
MP2	Lex program to identify the capital strings from the given input string.
MP3	Lex code for lexical analyser to eliminate white spaces & collecting no's from input data.
MP4	Lex program for simple Desktop calculator.
MP5	Lex program to print out all HTML Tags from the given file.

ITM(SLS) Baroda University
School of Computer Science, Engineering andTechnology
B.Tech - Semester III

Course Name-Database Management System
Course Code-C2310C4
Course Type: Core

Teaching Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
L	T	P	C	Theory Marks		Practical Marks		
3	0	4	5	External	Internal	External	Internal	
				40	60	0	50	150

Preamble-Database Management System or DBMS in short refers to the technology of storing and retrieving users' data with utmost efficiency along with appropriate security measures. It explains the basics of DBMS such as its architecture, data models, data schemas, data independence, E-R model, relation model, relational database design, and storage and file structure and much more.

What is DBMS?

Database Management System (DBMS) is softwarefor storing and retrieving user data while considering appropriate security measures. It consists of a group of programs which manipulate the database. The DBMS accepts the request for data from an application and instructs the operating system to provide the specific data. In large systems, a DBMS helps users and other third-party software to store and retrieve data.

DBMS allows users to create their own databases as per their requirement. The term "DBMS" includes the user of the database and other application programs. It provides an interface between the data and the software application.

Prerequisite: Basic knowledge of Computer Programming

Course Objective:

- To understand the different issues involved in the design and implementation of database system.
- To learn physical and logical database designs, database modelling, relational, hierarchical, and network models.
- To learn to use data manipulation language query, update, and manage a database. ● To understand essential DBMS concepts such as: database security, integrity, concurrency, storage strategies etc.

- The students will get the hands-on practice of using SQL and PL/SQL concepts.
- Design the database schema with the use of appropriate data types for storage of data in database.

Course Learning Outcome:

After completing the course, the student shall be able to :

	Course Outcome	**Bloom's Level**
CO 1	Understand functional components of the DBMS.	Understandin g
CO 2	Develop Data Models	Applying
CO 3	Design queries using Relational Algebra, Relational Calculus and SQL	Evaluation
CO 4	Design database schema	Creation
CO 5	Evaluate and optimize queries	Evaluation
CO 6	Understand transaction processing, concurrency control and recovery techniques	Understandin g
CO 7	Implementing the SQL Concepts	Evaluation

Course Competency:

1.Understand database concepts and structures and query language

2.Understand the E R model and relational model

3.To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

4.Understand Functional Dependency and Functional Decomposition. & apply various Normalization techniques

5.Perform PL/SQL programming using concept of Cursor Management, Error Handling, Package and Triggers

6.Execute various advance SQL queries related to Transaction Processing & Locking using concept of Concurrency control.

7.Understand query processing and techniques involved in query optimization.

Course Contents:

Unit #	**Topics (Programs) to be Completed**	**Hours**
1	**Database system architecture** Introduction to DBMS- Historical perspective, File Versus a DBMS, Advantages of DBMS, Describing and storing data in DBMS, Architecture of a DBMS	5

		Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML).	
	2	**Data models** Entity-relationship model, Features of ER Model, network model, relational and object oriented data models, integrity constraints, data manipulation operations.	6
	3	**Relational query languages** Relational algebra, Tuple and domain relational calculus and SQL – Queries, Constraints, Form of SQL Query , UNION, INERSECT and EXCEPT, Nested Queries, Aggregate Operators, Null values, Complex Integrity constraints in SQL, triggers and Embedded SQL	4
	4	**Relational database design** Domain and data dependency, Armstrong's axioms, Normal forms-1NF, 2NF, 3NF and BCN. Dependency preservation, Lossless design.	5
	5	**Query processing and optimization** Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms.	6
	6	**Transaction processing** Concurrency control, ACID property, Serializability of scheduling, Locking and time stamp based schedulers, Multi-version and optimistic Concurrency Control schemes, Database recovery. Security and Authorization- Access control, Direct access control and Mandatory access control, Role of DBA, Application development.	6
	7	**SQL Concepts** Basics of SQL, DDL,DML,DCL, structure – creation, alteration, defining constraints – Primary key, foreign key, unique, not null, check, IN operator, aggregate functions, Built-in functions –numeric, date, string functions, Set operations, sub-queries, correlated sub-queries, join, Exist, Any, All , view and its types., transaction control commands. **PL/SQL Concepts** : Cursors, Stored Procedures, Stored Function, Database Triggers	13
		Total	**45**

References:

Text Books :

1. Database System Concepts by Abraham Silberschatz, Henry F. Korth, S. Sudarshan – Tata McGraw Hill.

Reference Books:

1. Fundamentals of Database Systems by R.Elmasri and S. Navathe – Pearson.
2. An Introduction to Database Systems by C J Dave – Pearson.
3. Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems, 3nd Edition, McGraw Hill, 2003.
4. SQL- PL/SQL by Ivan bayross.

5. Oracle – The complete reference – TMH /oracle press

Required Software:

1.SQL/PLSQL Supporting Software-Oracle,SQLServer,MySql

Case Studies:

Sr.No.	**Case Studies**	**Evaluation**
C1	Railway Reservation – queries for maintaining database.	1. Identification of algorithm. 2. Report preparation. 3. Presentation with VIVA
C2	Facebook – ER diagram	1. Identification of algorithm. 2. Report preparation. 3. Presentation with VIVA
C3	Search button of facebook – uses which queries.	1. Identification of algorithm. 2. Report preparation. 3. Presentation with VIVA

Simulations/Animation: NA

TEDx Videos:

Sr. No.	**Videos**
T1	https://www.ted.com/talks/will_marshall_the_mission_to_create_a_searchable_database_ of_earth_s_surface - The mission to create a searchable database of Earth's surface – Will MarshalEarth is taken a huge database and using AI to index all the objects on the planet over time- which could make ships, trees, houses and everything else on Earth searchable, the same way you search Google. He shares a vision for how this database can become a living record of the immense physical changes happening across the globe. "You can't fix what you can't see," Marshall says. "

Other Videos:

Sr. No	**About Video**	**Link**	**Topic**
O1.	Stanford Lecture – Jennifer Widom - 05-Jan-2013 – Introduction to Databases	https://www.youtube.com/watch?v=D-kh0GuFmE&list=PLroEs25KGvwzmvIxYHRhoGTz9 w8LeXek0	Introduction to Database

O2.	NPTEL – Introduction to SQL/1 – By Prof. ParthaPratim Das – IIT Kharagpur	https://www.youtube.com/watch?v=w1XdPholz WY &list=PL3pGy4HtqwD3Ov1J2UBTfsLgxUzUkt TA M&index=7	SQL Concepts
O3.	NPTEL – Query Processing and Optimization - By Prof. ParthaPratim Das – IIT Kharagpur	https://www.youtube.com/watch?v=Sn_Wkf9KNEg	Query Processing and Optimization
O4.	CMU Database Systems - 01 Course introduction &Relational Data Model (Fall 2018) – Carnegie Mellon University	https://www.youtube.com/watch?v=vyVGm_2iFwU& list =PLSE8ODhjZXja3hgmuwhf89qboV1kOxMx7	Course introduction & Relational Data Model

Related MOOCs courses

1. Data Base Management System By Prof. ParthaPratim Das, Prof.Samiran Chattopadhyay - IIT Kharagpur – NPTEL- 8 weeks course

Activity Based Learning(ABL):

Sr.No	Name of Activity	Details of Activity	Outcome	Evaluation
A1	Describe Basics of PL/SQL	• Introduction to PL/SQL will be done. • Installations will be done.	Students will become familiar with PL/SQL language.	• Write queries • Get Output

A2.	Different types of Iteration and looping constructs in PL/SQL	● Inner and outer loops with examples. ● Use of EXIT statement ● Use of CONTINUE statement ● Use of GO TO statement	Students will be able to perform different looping examples. Able to write different queries.	● Write queries ● Get Output
A3.	Hands On Exercise on Cursor	● Learning above implicit and explicit cursors. ● Examples based on declaring, opening, fetching and closing a cursor	Students will be able write queries related to cursors.	● Write queries ● Get Output
A4.	Hands On Exercise on Stored Procedure	● Learning how to create a procedure. ● Examples based on procedures	Students will be able to create procedures and write queries.	● Write queries ● Get Output

Lab Experiments:

Sr. No.	**Experiments**	**Total Hrs**
PS-1	To study DDL-create and DML-insert commands. ● Create tables according to the following definition. ● Insert the data as shown below. ● From the above given tables perform the following queries: (1) Describe deposit, branch. (2) Describe borrow, customers. (3) List all data from table DEPOSIT. (4) List all data from table BORROW. (5) List all data from table CUSTOMERS. (6) List all data from table BRANCH. (7) Give account no and amount of depositors. (8) Give name of depositors having amount greater than 4000. (9) Give the name of customers who opened account after date ‘1-12-96’.	2

PS-2	Create the below given table and insert the data accordingly. Perform following queries. (10) Retrieve all data from employee, jobs and deposit. (11) Give details of account no. and deposited rupees of customers having opened account between dates 01-01-06 and 25-7-06. (12) Display all jobs with minimum salary is greater than 4000. (13) Display names and salary of employee whose department no is 20. Give alias name to name of employee. (14) Display employee no, name and department details of those employee whose department lies in (10,20). To study various options of LIKE predicate. (1) Display all employee whose name start with 'A' and third character is 'a'. (2) Display name, number and salary of those employees whose name is 5 characters long and first three characters are 'Ani'. (3) Display the non-null values of employees and also employee name second character should be 'n' and string should be 5 character long. (4) Display the null values of employee and also employee name's third character should be 'a'. (5) What will be output if you are giving LIKE predicate as '%_%'ESCAPE'\'.	2
PS-3	To perform various data manipulation commands, aggregate functions and sorting concept on all created tables. (1) List total deposit from deposit. (2) List total loan from karolbagh branch (3) Give maximum loan from branch vrce. (4) Count total number of customers. (5) Count total number of customer's cities. (6) Create table supplier from employee with all the columns. (7) Create table sup1 from employee with first two columns. (8) Create table sup2 from employee with no data. (9) Insert the data into sup2 from employee whose second character should be 'n' and string should be 5 characters long in employee name field. (10) Delete all the rows from sup1. (11) Delete the detail of supplier whose sup_no is 103. (12) Rename the table sup2. (13) Destroy table sup1 with all the data. (14) Update the value dept_no to 10 where second character of emp. name is 'm'. (15) Update the value of employee name whose employee number is 103.	2

PS-4	To study Single-row functions. (1) Write a query to display the current date. (2) For each employee, display the employee number, job, salary, and salary increased by 15% and expressed as a whole number. Label the column New Salary. (3) Modify your query no 4.(2) to add a column that subtracts the old salary from the new salary. Label the column Increase. (4) Write a query that displays the employee's names with the first letter capitalized and all other letters lowercase, and the length of the names, for all employees whose name starts with J, A, or M. Give each column an appropriate label. Sort the results by the employees' last names. (5) Write a query that produces the following for each employee: <employee last name> earns <salary> monthly. (6) Display the name, hire date, number of months employed and day of the week on which the employee has started. Order the results by the day of the week starting with Monday. (7) Display the hiredate of emp in a format that appears as Seventh of June 1994 12:00:00 AM. (8) Write a query to calculate the annual compensation of all employees (sal+comm.).	2
PS-5	2. Displaying data from Multiple Tables (join) (1) Give details of customers ANIL. (2) Give name of customer who are borrowers and depositors and having living city Nagpur (3) Give city as their city name of customers having same living branch. (4) Write a query to display the last name, department number, and department name for all employees. (5) Create a unique listing of all jobs that are in department 30. Include the location of the department in the output (6) Write a query to display the employee name, department number, and department name for all employees who work in NEW YORK. (7) Display the employee last name and employee number along with their manager's last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively. (8) Create a query to display the name and hire date of any employee hired after employee SCOTT.	3.

PS-6	4. To apply the concept of Aggregating Data using Group functions. (1) List total deposit of customer having account date after 1-jan-96. (2) List total deposit of customers living in city Nagpur. (3) List maximum deposit of customers living in bombay. (4) Display the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number. (5) Write a query that displays the difference between the highest and lowest salaries. Label the column DIFFERENCE. (6) Create a query that will display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998 (7) Find the average salaries for each department without displaying the respective department numbers. (8) Write a query to display the total salary being paid to each job title, within each department. (9) Find the average salaries > 2000 for each department without displaying the respective department numbers. (10)Display the job and total salary for each job with a total salary amount exceeding 3000, in which excludes president and sorts the list by the total salary. (11)List the branches having sum of deposit more than 5000 and located in city bombay.	2
PS-7	5. To solve queries using the concept of sub query. (1) Write a query to display the last name and hire date of any employee in the same department as SCOTT. Exclude SCOTT (2) Give name of customers who are depositors having same branch city of mr. sunil. (3) Give deposit details and loan details of customer in same city where pramod is living. (4) Create a query to display the employee numbers and last names of all employees who earn more than the average salary. Sort the results in ascending order of salary. (5) Give names of depositors having same living city as mr. anil and having deposit amount greater than 2000 (6) Display the last name and salary of every employee who reports to ford. (7) Display the department number, name, and job for every employee in the Accounting department. (8) List the name of branch having highest number of depositors. (9) Give the name of cities where in which the maximum numbers of branches are located. (10) Give name of customers living in same city where maximum depositors are located.	2
PS-8	6. Manipulating Data (1) Give 10% interest to all depositors. (2) Give 10% interest to all depositors having branch vrce (3) Give 10% interest to all depositors living in nagpur and having branch city bombay.	2

	(4) Write a query which changes the department number of all employees with empno 7788’s job to employee 7844’current department number. (5) Transfer 10 Rs from account of anil to sunil if both are having same branch. (6) Give 100 Rs more to all depositors if they are maximum depositors in their respective branch. (7) Delete depositors of branches having number of customers between 1 to 3. (8) Delete deposit of vijay. (9) Delete borrower of branches having average loan less than 1000.	
PS-9	To apply the concept of security and privileges.	1
PS-10	To study transaction control commands.	1
PS-11	Write cursors and triggers.	1

Mini Projects:

Sr.no.	
MP-1	Inventory Control Management Database
MP-2	Student Record Keeping System Database
MP-3	College Student Database System
MP-4	Library Management System
MP-5	Payroll Management Database System

ITM (SLS) Baroda University

School of Computer Science, Engineering and Technology

B.Tech - Semester IV

Course Name: Data Structures and Algorithms-II

Course Code: C2410C1

Course Type: Core

Teaching Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
L	T	P	C	Theory Marks		Practical Marks		
				External	Internal	External	Internal	
4	0	2	5	40	60	0	50	150

Preamble:

This course has three components - advanced data structures, algorithm Efficiency and Algorithm correctness via Proof Rules (Axiomatic Semantics) based design. This course focuses on the modern theory of algorithms, the themes of efficient algorithms and intractable problems.

Course Overview:

In this course, students will learn concepts of augmented list structures, graphs, dynamic programming applications, efficiency of algorithms, and techniques to evaluate the correctness of algorithms using design rules.

Prerequisite:

Fundamental concepts of Programming,data structures and Algorithms

Course Learning Outcomes:

After completing the course, the student shall be able to:

	Course Outcome	Bloom's Level
CO1	Selection and analysis of algorithms, including run time analysis.	Applying
CO2	Understand the concept of Time Complexity and compute it through order notations for a given problem.	Understanding andApplying

CO3	Develop, analyze and implement algorithms for augmented list structures.	Applying
CO4	Develop, analyze and implement algorithms for non-linear algorithms like graphs,trees and hash tables.	Applying
CO5	Implement applications using dynamic programming along withSpecified mini project.	Applying

Competencies:

After completion of the this course, our students shall acquire the following genericcompetencies:

1 Implement greedy algorithms, divide and conquer , sorting and Dynamic programming to solve programming challenges.
2 Apply graph algorithms to solve real-world challenges: finding shortest paths on maps.
3 Design and Implement algorithms for programming challenges using techniques: collision resolution, separate chaining, Van Emde Boas tree, AVL, Red-Black, B tree, Huffman tree, Skip-lists.

Course Syllabus:

Unit #	Topics	Teaching Hours
1	**Advanced Data Structures-Trees and Hash Tables:** 1.1. Trees - Splay, Trie, AVL Tree, Red-Black Tree, 2-3, B-tree, Huffman tree, Skip-lists 1.2 Hash Tables: hash functions, collision resolution, separatechaining 1.3 Patterns and String Matching: Time-Space trade-off, Naive,Rabin Karp, Boyer-Moore Algorithms	10
2	**Advanced Data Structures- Augmented List Structure andGraphs** 2.1.Augmented List Structures: interval overlap and interval tree 2.2Graphs: minimum spanning tree, Prim's, Kruskal algorithms,Union Find data structure, Disjoint sets and quick-union algorithms, Dijkstra's shortest path algorithm. 2.3 **Dynamic programming application examples:** rod cutting,multi stage graphs, Travelling Salesman, Longest Common Subsequence (LCS), shortest paths, maximum flow problems	15

3	**Algorithm Efficiency and Complexity:** 3.1.Efficiency of algorithms: loops, redundant computations, late termination, Early Detection of termination. 3.2. Estimating the running times: problem size as a measure, 3.3 Order notations: Big-Oh, Theta, Omega, 3.4. Solution ofsimple recurrence relations 3.5. Computation models and Design by Refinement – Functional Model: Features of Functional Model, Recursive Processes, Analysis of Correctness and Efficiency, Imperative Model: Primitives for the Imperative Model, Specifications and Prototyping, Examples of Step wise Refinement.	13
4	**Algorithm Correctness and design using Proof Rules:** 4.1 Assertions at the input and theoutput of a computation block 4.2 Proof Rules: Compound Statement, Conditional Statements, Case Statements, Repetitive Statements, Repeat Until Statement, Example: Division Algorithm 4.2. Correct Termination of algorithm	9
5	**Algorithm Correctness and design using Proof Rules-Advanced Constructs:** 5.1Proof Rules for advanced constructs like FOR loops 5.2Design by proof rules: Referring to previous values of variables, the interface specifications	8
	TOTAL	**55**

ITM BARODA UNIVERSITY Vadodara | Gujarat | India **Reference :**

Textbooks:

1. P.H.Dave and H.B.Dave, "*Design and Analysis of Algorithms*", Pearson Educ.2nd Edition.
2. Cormen, Leiserson, Rivest and Stein, "Introduction to algorithms ",The MIT Press. 3rdEdition.

Reference Books:

1. Steven Skiena, "The Algorithm Design Manual", Springer-Verlag.
2. Kleinberg and Eva Tardos, "Algorithm Design", Pearson Publication.
3. A.VAho, J. E. Hopcroft, and J. D. Ullman:, "Data Structures and Algorithms" ,Addison‑Wesley.

TedEx Videos:

Sr. No	**TEDx Video**

T1	How algorithms shape our world.- Kevin Slavin https://www.youtube.com/watch?v=ENWVRcMGDoU Kevin Slavin argues that we're living in a world designed for -- and increasingly controlled by -- algorithms. In this riveting talk from TEDGlobal, he shows how these complex computer programs determine espionage tactics, stock prices, movie scripts, and architecture. Slavin also warns that we are writing code we can't understand with implications we can't control.
T2	What's an algorithm - David J. Malan https://www.youtube.com/watch?v=6hfOvs8pY1k - An algorithm is a method of solving problems both big and small. Though computers run algorithms constantly, humans can also solve problems with algorithms. David J. Malan explains how algorithms can be used in seemingly simple situations and also complex ones.

Other Videos:

Sr. No	**About Video**	**Link**	**Topic**
O1	Lecture Series on Data Structures and Algorithms by Dr. Naveen Garg, Department of Computer Science and Engineering ,IIT Delhi.	https://www.youtube.com/watch?v =JRsN4Oz36QU	Red black Trees
O2	Lecture on Asymptotic Complexity , Peak finding by Prof. Victor Costan, MIT Open Course ware	https://www.youtube.com/watch?v =P7frcB_-g4w	Asymptotic Complexity, Peak Finding
O3	Lecture by ProfessorMadhavan Mukund, Chennai Mathematical Institute.	https://www.youtube.com/watch?v =P-oDU7AQ8Mo	Longest Common Subsequence
O4	Lecture by Professor Madhavan Mukund, Chennai Mathematical Institute.	https://www.youtube.com/watch?v =EF953Vafi5c	Introduction to Dynamic Programming

Related MOOCs courses:

Sr.No	**MOOC Courses**

M1	Programming, Data Structures And Algorithms Using Python by Prof. Madhavan Mukund 8 Weeks on NPTEL.
M2	Data Structures and Algorithms Specialization Offered By University ofCalifornia San Diego National Research University Higher School of Economics(Coursera)

Lab Experiments:

Sr. No.	Program Statement	Appr. Hours
P1	Write a python code to demonstrate basic operations such asinsertion,look-up and removal on splay tree.	4
P2	Implement a python program for performing following operations on VanEmde Boas tree. · *Insert*: insert a key/value pair with an *m*-bit key · *Delete*: remove the key/value pair with a given key · *Lookup*: find the value associated with a given key · *FindNext*: find the key/value pair with the smallest key whichis greater than a given *k* · *FindPrevious*: find the key/value pair with the largest keywhich is smaller than a given *k*	8
P3	Write a python program for inserting an element into a Red-Black Tree.	4
P4	Write a python program for deleting an element into a Red-Black Tree.	4
P5	Writea python program to implement hash table with separatechaining.	4
P6	Write a python code for Prim's and Kruskal's Minimum Spanning Tree algorithms.	4
P7	Write python code to compute order notations for merge sort, bubble sort and insertion sort.	4

P8	Write python code to perform unit testing using assertln() function as well as to check loops, conditional statements.	4
P9	Write python program for maximum flow problem.	4

Mini Project:

Apart from the above laboratory work, each student will implement a teacher specified individual small project requiring implementation through more than one source file and one header file, algorithm selection, determining the test data sets and a documentation for a 10-15 minutes presentation of his/her work. Some of the examples are listed below :

MP1	Rod Cutting Problem You have given a rod of length n and an array of prices that contains prices of all pieces of the size which are smaller than n. You need to determine the maximum value obtainable by cutting up the rod and selling its pieces.
MP2	Use Dynamic programming to solve Travelling Sales Man ProblemInput: A Complete graph with weights on edges and budget b. Output: A cycle that visits each vertex exactly once and has total weight at most b.

ITM(SLS) Baroda University
School of Computer Science, Engineering and Technology
B.Tech CSE-CSN - Semester - IV

Course Name: Cryptography and Network Security
Course Code: C2430C1
Course Type: Core (For CSE-CSN)

Teaching Scheme:

Teaching Scheme			**Credits**	**Examination Marks**				**Total Marks**
L	**T**	**P**	**C**	**Theory Marks**		**Practical Marks**		
4	**0**	**2**	**5**	**External**	**Internal**	**External**	**Internal**	
				40%	**60%**	**0**	**50**	**150**

Course Competencies:

1. Understand various Network Security Objectives and their corresponding Mechanisms.
2. Implementation of various cryptographic algorithms and authentication algorithms.

Course Objective:

- Expose the students to various network security objectives and their corresponding mechanisms.
- Understand the working of X.509 based Public Key Infrastructure
- Understand the working of SSL/TLS

Course Learning Outcomes:
At the end of the course the student will be able to:

	COURSE OUTCOME	**BLOOM'S LEVEL**
CO1	Understand various network security objectives and mechanisms	Understanding
CO2	Analyze which network security objectives would be required in a given application scenario	Analyzing
CO3	Design secure networked applications	Analyzing
CO4	Implement secure networked applications	Applying
CO5	Understand working of X.509 based PKI	Understanding

Theory Syllabus:

Sr. No.	Content	% Weightage
01	Introduction – security services, security services, security mechanisms, modular arithmetic, TheEuclidean algorithm.	15%
02	Symmetric Cipher Model, Cryptography, Cryptanalysis and Attacks; Substitution and Transposition techniques	10%
03	Stream ciphers and block ciphers, Block Cipher structure, Data Encryption standard (DES) with example, strength of DES, Design principles of block cipher, Introduction to AES and its operations with structure, its transformation functions, key expansion, example and implementation	10%
04	Multiple encryption and triple DES, Electronic Code Book, Cipher Block Chaining Mode, Cipher Feedback mode, Output Feedback mode, Counter mode	5%
05	Public Key Cryptosystems with Applications, Requirements and Cryptanalysis, RSA algorithm, its computational aspects and security, Diffie-Hillman Key Exchange algorithm, Man-in-Middle attack	15%
06	Cryptographic Hash Functions, their applications, Simple hash functions, its requirements and security, Secure Hash Algorithm (SHA)	10%
07	Message Authentication Codes, its requirements and security, MACs based on Hash Functions, HMAC	10%
08	Digital Signature, its properties, requirements and security, NIST digital Signature algorithm	8%
09	Key management and distribution, symmetric key distribution using symmetric and asymmetric encryptions, distribution of public keys, X.509 certificates, Public key infrastructure	7%
10	Web Security threats and approaches, SSL architecture and protocol, Transport layer security, HTTPS	10%

Main Text-Book:

1. Cryptography and Network Security: Principles and Practice, William Stallings, 6th Edition, Pearson Publishers

Reference Books:

2. Cryptography and Network Security, Behrouz Forouzan, McGraw Hill Publications

TOPIC-WISE THEORY SYLLABUS FROM TEXT-BOOK

Chapter #	Topic #
1	1.3, 1.4, 1.5
2	2.1, 2.2, 2.3
3	3.1, 3.2, 3.3, 3.4, 3.5
4	4.1, 4.2, 4.3
5	5.2
6	6.1, 6.2, 6.3, 6.4, 6.5, 6.6
9	9.1, 9.2
10	10.1
11	11.1, 11.2, 11.3, 11.5
12	12.1, 12.2, 12.3, 12.4, 12.5
13	13.1, 13.4
14	14.1, 14.2, 14.3, 14.4, 14.5
17	17.1, 17.2, 17.3, 17.4

List of Practical/Laboratory work: (Programs may be implemented in C/Python/Java/etc.)

Sr.#	Experiment/Practical
1.	Implement a Program to demonstrate the working of generalized Caesar cipher.
2.	Implement a Program to demonstrate the working of a Brute Force Attack.
3.	Implement a Program to demonstrate the working of mono-alphabetic substitution cipher.
4.	Implement a Program to demonstrate the working of transposition cipher.
5.	Use the tools4noobs tool to understand the terminal behavior of various cryptographic algorithms. Message Digests, etc.
6.	Implement a Program to demonstrate the working of DES.
7	Implement a Program to demonstrate the working of Triple DES.
8	Implement a Program to demonstrate the working of AES.
9	Implement a Program to demonstrate the working of MD5.
10	Implement a Program to demonstrate the working of SHA-1.
11	Implement a Program to demonstrate the working of HMAC.
12	Implement a Program to demonstrate the working of RSA.
13	Implement a Program to generate random numbers using any two popular PRNG algorithms.
14	Implement a Program to demonstrate the working of Diffie Hellman Key Exchange Algorithm.
15	Implement a program to classify a given password as weak or strong based upon pre-defined password policy.

Case Studies:

Sr.No	Case Studies	Evaluation
C1	Understand the working of Controller of Certifying Authority India cca.gov.in	1. Report preparation. 2. Presentation with VIVA

C2	Understand the working of the CA e-mudhra by visiting its website	1. Report preparation. 2. Presentation with VIVA

Web References/MOOC resources:

1. https://onlinecourses.nptel.ac.in/noc21_cs16/preview
   NPTEL Cryptography and Network Security Course

2. https://nptel.ac.in/courses/106106129
   NPTEL Cryptography and Network Security Course

3. http://williamstallings.com/Cryptography/
   Text Book Author William Stallings online resources portal for cryptography and network security

ITM (SLS) Baroda University
School of Computer Science, Engineering and Technology
B.Tech – Semester IV

Course Name: Operating System
Course Code: C2410C3
Course Type: Core

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
L	T	P	C	Theory Marks		Practical Marks		
				External	Internal	External	Internal	
4	0	2	5	40	60	0	50	150

Preamble:
This course is concerned with the principles and concepts that govern the design of modern computer operating systems are studied. Managing computing resources such as the memory, the processor and the Input/output devices are covered. Algorithms for CPU scheduling, memory and general resource allocation; process coordination and management; deadlocks and memory management techniques; case studies of Linux operating systems are also covered.

What is Operating System?
Operating System Subject provides the basic and advanced concepts of operating system. Operating System can be defined as an interface between user and the hardware. It provides an environment to the user so that, the user can perform its task in convenient and efficient way.An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.

Prerequisites: Data structures, working experience of any one structured programming language

Course Learning Outcome:
After completing the course, the student shall be able to:

	Course Outcome	Bloom's Level
CO1	Analyze the structure of OS and basic architectural components involved in OS design.	Analyze
CO2	Understand the concept of a process and thread and apply the cons of process/thread scheduling.	Understanding
CO3	Apply the concept of process synchronization, mutual exclusion and	Applying

	Deadlock in contemporary operating system.	
CO4	Analyzethe various memory management techniques and Apply various algorithms for memory management.	Analyze
CO5	Understand the I/O management and security aspects of operating system	Understanding
CO6	Implement shell scripts in Unix/Linux O.S and understand virtualization concept.	Applying

Course Syllabus:

Unit #	***Topics (Programs) to be Completed***	***Total Hrs.***
1	**Overview of operating system**: Computer system overview, Architecture, Goals & Structures of O.S, Operating System Objectives and Basic functions, Interaction of O.S. & hardware architecture, System calls. **Types of OS**: Batch, multiprogramming. Multitasking, Time-sharing, Parallel, Distributed &Realtime O.S.	04
2	**Process description and control** **Process:** Concept of a Process, Process States, Process Description, Process Control (Process creation, waiting for the process/processes, Loading programs into processes and Process Termination), Execution of the Operating System. **Threads:** Processes and Threads, Concept of Multithreading, Types of Threads, Thread programming Using Pthreads, Thread scheduling, System calls like ps, fork, join, exec family, wait. **Scheduling:** Types of scheduling: Pre-emptive, Non pre-emptive, Scheduling algorithms	08
3	**Concurrency control and Inter-process communication** **IPC**: Pipes, shared memory, message passing. **Process/thread Synchronization and Mutual Exclusion:** Principles of Concurrency, Requirements for Mutual Exclusion, Race Conditions, Critical Section, Peterson’s Solution, Mutual Exclusion: Hardware Support, Operating System Support (Semaphores and Mutex), Programming Language Support (Monitors). **Classical synchronization problems**: Readers/Writers Problem, Producer and Consumer problem, Dinning Philosopher Problem.	10
4	**Deadlock:** Principles of Deadlock, Deadlock Modelling, Strategies to deal with deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock detection and recovery, Starvation, System calls like signal, kill, Priority Inversion	4

5	**Memory management:** **Memory Management:**Memory Management Requirements, Memory Partitioning: Fixed and Variable Partitioning, Swapping, Memory Allocation Strategies (First Fit, Best Fit, and Worst Fit), Swapping, Paging and Fragmentation. Page Table Structure, Segmentation, Demand Paging, Buddy System, Security Issues, Concept of north bridge and south bridge. **Virtual Memory**: Concepts, Hardware and Control Structures, Page Replacement Policies (FIFO, LRU, Optimal, Other Strategies), Thrashing.	07
6	**I/O management & Disk scheduling and Security & Protection:** **I/O management**: I/O Devices, Organization of I/O functions, Operating System Design issues, I/O Buffering, *File management* **DiskScheduling** (FCFS, SCAN, C-SCAN, SSTF), RAID, Disk Cache. **Security & Protection:** Security Environment, Design Principles of Security, User Authentication, Protection Mechanism: Protection Domain, Access Control List	07
7	**Unix/Linux Operating System:** Design Principles of Unix/Linux, Role & Function of Kernel, Linux Booting Process, System Calls, Directory Structure, System Administration, ElementaryLinux command. **Shell Programming**: BASH Shell scripting: Basic shell commands, shell as a scripting language. **Virtual machines:** supporting multiple operating systems simultaneously on a single hardware platform; running one operating system on top of another. True or pure virtualization	05
	Total	**45**

Reference :

Textbooks

Sr. No Book Name

1 William Stallings, Operating System: Internals and Design Principles, Prentice Hall, ISBN-10: 0-13-380591-3, ISBN-13: 978-0-13-380591-8, 8th Edition

2 Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, “Operating System Concepts”, WILEY, ISBN 978-1-118-06333-0 , 9th Edition

3 Andrew S. Tanenbaum& Herbert Bos, “Modern Operating System”, Pearson, ISBN-13: 9780133592221, 4th Edition

Reference Books:

Sr. No Book Name

1 William Stallings, “Operating Systems: Internals & Design Principles”, 9th Edition, Pearson Education India

2 Peter B. Galvin, Greg Gagne, Abraham Silberschatz, “Operating System Concepts”, 9th edition, John Wiley & Sons, Inc.

3 Andrew S. Tanenbaum,“Modern Operating Systems”.

Required Software: LINUX OS, C compiler

Simulations/Animation: https://ess.cs.tu-dortmund.de/Software/AnimOS/CPU-Scheduling/

Case Studies:

	Case Studies	**Evaluation**
C1.	KALI LINUX Operating System	1. Report preparation. **2.** Presentation with VIVA
C2.	Xen Family – a virtual machine monitor	
C3.	Influential Operating Systems	

Simulations/Animation: https://ess.cs.tu-dortmund.de/Software/AnimOS/CPU-Scheduling/

Other Videos Links:

Sr. No	**About Video**	**Link**	**Topic**
O1	UC Berkeley Computer Science 162, 001 – Fall 2010 Operating Systems and System Programming Instructor: John Kubiatowicz	https://youtu.be/tvPv7gLcwpc	Operating System structure
O2	University of California, Berkeley: Computer Science 162, 001 – Fall 2010“Concurrency: Processes, Threads, and Address Spaces”Instructor: John Kubiatowicz	https://youtu.be/5Ip__MiHRxw	Concurrency: Process, Thread and address space

O3	University of California, Berkeley: John Kubiatowicz's"Mutual Exclusion, Semaphores, Monitors and Condition Variables"	https://youtu.be/P19nmWurCz8	Mutual Exclusion, Semaphores, Monitors and Condition Variables"
O4	Massachusetts Institute of Technology: Hari Balakrishnan and Samuel Madden's "Virtual Memory"	https://youtu.be/TPeKtQykujI	Virtual Memory

O5	IIT Madras- Introduction to operating system- Deadlock, Prof Chester Rebeiro.	https://youtu.be/UczJ7misUEk	Deadlocks
O6	Linux Programming & Scripting by Prof.AnandIyer,Director, Calypto Design Systems.	https://youtu.be/akU1Ji8Vzdk	Linux Basics

Related MOOCs courses :

Sr. No	MOOC Courses
M1	Operating System and you Becoming a power User Offered by Google(Coursera)
M2	Introduction to Operating Systems by Prof. Chester Rebeiro 08 weeks on NPTEL. For More details on NPTEL visit http://nptel.iitm.ac.in

Lab Experiments:

PS1	Study of Basic commands of Linux/UNIX.
PS2	Study of Advance commands and filters of Linux/UNIX.
PS3	Shell programming: Write a program to implement an address book with options given below: a) Create address book. B) View address book. C) Insert a record. D) Delete a record. e) Modify a record. F) Exit.
PS4	Write a shell script to generate marksheet of a student. Take 3 subjects, calculate and display total marks, percentage and Class obtained by the student

PS5	Write a shell script to a) display multiplication table of given number b) find factorial of given number n.
PS6	Write a shell script which a) will accept a number b and display first n prime numbers as output. b) will generate first n Fibonacci numbers like: 1, 1, 2, 3, 5, 13, … c) will check entered string is palindrome or not.

PS7	Write a menu driven shell script which will print the following menu and execute the given task a) Display calendar of current month b) Display today's date and time and a welcome message (like Good Morning etc.). The time should be displayed with "a.m." or "p.m." and not in 24 hours notation. c) Display usernames those are currently logged in the system d) Display your name at given x, y position e) Display your terminal number
PS8	Write a shell script to read n numbers as command arguments and sort them in descending order.
PS9	Write a shell script f) To find the global complete path for any file. g) To broadcast a message to a specified user or a group of users logged on any terminal. h) To copy the file system from two directories to a new directory in such a way that only the latest file is copied in case there are common files in both the directories.
PS10	Write a shell script a) To compare identically named files in two different directoriesand if they are same, copy one of them in a third directory. b) To display the name of those files (in the given directory) which are having multiple links. c) Write a script to display the directory in the descending order of the size of each file.

PS11	Write a shell script to display all executable files, directories and zero sized files from current directory.
PS12	Write an awk program using function, which convert each word in a given text into capital.
PS13	Write a shell script to validate the entered date. (e.g. Date format is : dd-mm-yyyy).
PS14	Understand the concept of virtual Machine and installation of virtual machine using VM Box /VMware
PS15	Write a program for **a)** Implementation of FCFS (First Come First Serve) CPU Scheduling. **b)** Implementation of SJF (Shortest Job First) CPU Scheduling. **c)** Implementation of Round Robin (RR) CPU Scheduling. **d)** Implementation of Priority CPU Scheduling Algorithm.
PS16	Write a program for a) Implementation of FIFO Replacement Algorithm b) Implementation of Optimal Page Replacement Algorithm. c) Implementation of LRU Page Replacement Algorithm by Stack method.
PS17	Write a program to implement (at least one) deadlock avoidance techniques

ITM(SLS) Baroda University
School of Computer Science, Engineering and Technology
B.Tech - Semester – IV (CSE/IT/AI/CSN)

Course Name: Computer Networking
Course Code: C2410C4
Course Type: Core

Teaching Scheme:

Teaching Scheme			**Credits**	**Examination Marks**				**Total Marks**
L	**T**	**P**	**C**	**Theory Marks**		**Practical Marks**		
4	**0**	**2**	**5**	**External**	**Internal**	**External**	**Internal**	
				40%	**60%**	**0**	**50**	**150**

Competencies:

1. Implement different layers of networks.
2. Implementation of remote procedure calls.

Course Objective: Objective includes

- Learning about computer network organization and implementation,
- Understanding of data communication and computer networks
- Installation, monitoring, and troubleshooting of current LAN systems.

Course Learning Outcomes:
At the end of the course the student will be able to:

	COURSE OUTCOME	**BLOOM'S LEVEL**
CO1	Understand OSI and TCP/IP models	Understanding
CO2	Analyze MAC layer protocols and LAN technologies	Analyzing
CO3	Design applications using internet protocols	Analyzing
CO4	Implement routing and congestion control algorithms	Analyzing
CO5	Develop application layer protocols	Applying

Course syllabus:

Unit	Course Content	Weightage percentage
Unit I	**Introduction:** Uses of Computer Networks, Personal Area Network, Local Area Networks, Metropolitan Area Networks, Wide Area Networks, Internetworks, Network software, protocol hierarchies, Design issues for the layers, connection oriented vs. Connectionless service, service primitives, relationship of services in protocols, Reference Models, Open System Interconnection (OSI), TCP/IP Reference models	**15%**
Unit II	**Physical Layer** Guided Transmission Media, Magnetic Media, Twisted Pairs, Coaxial Cable, Power Lines, Fiber Optics, Wireless Transmission, Electromagnetic Spectrum, Radio Transmission, Microwave Transmission, Infrared Transmission,	**15%**
Unit III	**Data Link Layer** Design issues, Error detection and correction. Elementary data link protocols: Utopian simplex protocol, a simplex stop and wait protocol for an error-free channel, a simplex stop and wait protocol for noisy channel. Sliding Window protocols	**15%**
Unit IV	**Medium Access Control Sublayer** The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols (CSMA/CD and CSMA/CA), MAC issues in Wireless Networks (Hidden Station and Exposed Station Problem, RTS/CTS scheme), Wireless LAN (IEEE 802.11), Bluetooth, Data Link Layer Switching	**10%**
Unit V	**Network Layer** Design issues, Routing algorithms: Optimality principle, shortest path routing, Flooding, distance vector routing, Link State routing, Congestion Control Algorithms.	**15%**

Unit VI	**Transport Layer:** Transport Services, Elements of Transport protocols, Connection establishment, connection release, Error control, flow control, congestion control, UDP and TCP protocols.	**15%**
Unit VII	**Application Layer:** Domain name system, Electronic Mail, HTTP	**15%**

Topic-Wise Syllabus from Text-Book (Text-Book Name mentioned below):

Unit	Topics
I	Chapter 1: 1.1.1, 1.2, 1.3, 1.4
II	Chapter 2: 2.2.1, 2.2.2, 2.2.3, 2.2.5
III	Chapter 3: 3.1, 3.2, 3.3, 3.4
IV	Chapter 4: 4.1, 4.2.1, 4.2.2, 4.2.5, 4.3.2, 4.3.4, 4.4.1, 4.4.4, 4.4.5, 4.6.1, 4.6.2, 4.8.4
V	Chapter 5: 5.1, 5.2.1, 5.2.2, 5.2.3, 5.2.4, 5.2.5, 5.3
VI	Chapter 6: 6.1.1, 6.1.2, 6.2.1, 6.2.2, 6.2.3, 6.4.1, 6.5.1, 6.5.2, 6.5.3, 6.5.4, 6.5.5, 6.5.6
VII	Chapter 7: 7.1, 7.2, 7.3.4

Text Book:

1. Andrew S. Tanenbaum, David J Wetherall, “Computer Networks”, 5th Edition, Pearson Edu, 2010.

Reference Books:

1. Kurose and Ross,”Computer Networking- A Top-Down approach”, 6th edition, Pearson
2. M.L.Liu,”Distributed Computing: Principles and Applications”,Pearson
3. Fred Halsall, Addison Wesley,”Computer Networking and the Internet”, 5th edition
4. Behrouz Forouzan,”TCP/IP Protocol Suite”,4th edition, McGraw Hill
5. Behrouz Forouzan,”Data Communications and Networking “,5th edition, McGraw Hill

Case Studies:

Sr.No	**Case Studies**	**Evaluation**
C1	How to provide the security in a network.	1. Report preparation. 2. Presentation with

C2	A message is to be transmitted from sender to receiver using TCP, so how it is going to be transmitted at each layer.	VIVA

TEDx Videos:

Sr.No	**TEDx Video**
T1	https://www.youtube.com/watch?v=Cj98mr_wUA0 - An introvert's guide to networking | Rick Turoczy Rick explains that every person knows someone that someone else should know. That introverts can be comfortable with being uncomfortable by "collecting dots" and connecting dots only that willing individuals can see. If we are intentional on our own terms we can build community by making common sense, common.
T2	https://www.youtube.com/watch?v=IjSPfGsaC3g- The art of active networking - Mark E. Sackett ,This talk was given at a local TEDx event, produced independently of the TED Conferences. Four simple questions are the start of a better way to meet people. So many people of us meet one another, collect a stack of business cards, and then throw them out without ever truly connecting. After bringing together hundreds of thousands of people, Mark shares his insights and outlines a way for us to make the most of those whose paths we cross.

Other Videos:

Sr. No	**About Video**	**Link**	**Topic**
01	Prof SoumyaKanti Ghosh & Prof Sandip Chakraborty - Talking about the networking	https://www.youtube.com/watch?v=O--rkQNKqls&list=PLbRMhDVUMngf-peFloB7kyiA40EptH1up	Introduction to computer network.
02	MIT 6.858 Computer Systems Security	https://www.youtube.com/watch?v=GqmQg-cszw4&list=PLUl4u3cNGP62K2DjQLRxDNRi0z2IRWnNh	Computer security
03	CS144 Fall 2013 by Stanford University Sharjeel Sayed	https://www.youtube.com/watch?v=nh970YyKRDA&list=PLvFG2xYBrYAQCyz4Wx3NPoYJOFjvU7g2Z	Networks and congestion control

MOOCs courses

Sr.No	MOOC Course
M1	Computer Networks by Prof. Sujoy Ghosh IIT Kharagpur on nptel
M2	Computer Networks and Internet Protocol by IIT Kharagpur on nptel
M3	Network simulation using NS2 by Durgesh P. Kshirsagar on udemy

Lab Experiments:

Sr. No	Experiments	Hrs.
P1	Study of different network devices in detail.	2
P2	Study of different types of network cables and practically implement the cross-wired cable and straight through cable using clamping tool.	4
P3	Study of basic network command and Network configuration commands	2
P4	Implement different LAN topologies using Network Simulator.	2
P5	Implement the concept of VLAN using Network Simulator	2
P6	Implement the concept of static routing.	2
P7	Implement the concept of dynamic routing (RIP, OSPF, BGP).	2
P8	Packet capture and header analysis by wire-shark (TCP,UDP,IP)	4
P9	Installation, Configuration and Operations of Apache Web Server	2
P10	Implement an Intranet based Static Web Site using Apache Web Server	2
P11	Host a static web site publicly on the Internet using Free Web Hosting	2

Mini Projects

Sr.No	Mini Projects
MP1.	Send and Receive Mail using Mozilla Thunderbird to understand working of Non Web Based Emailing System
MP2	Use Telnet to Send Email
MP3	Installation, Configuration and Operation of FileZilla FTP client and server