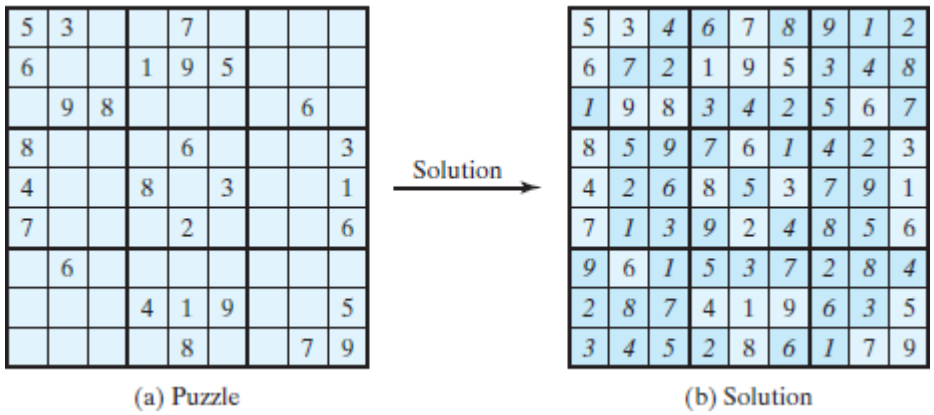
CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY
FACULTY OF TECHNOLOGY AND ENGINEERING
Devang Patel Institute of Advance Technology and Research
CE/CSE

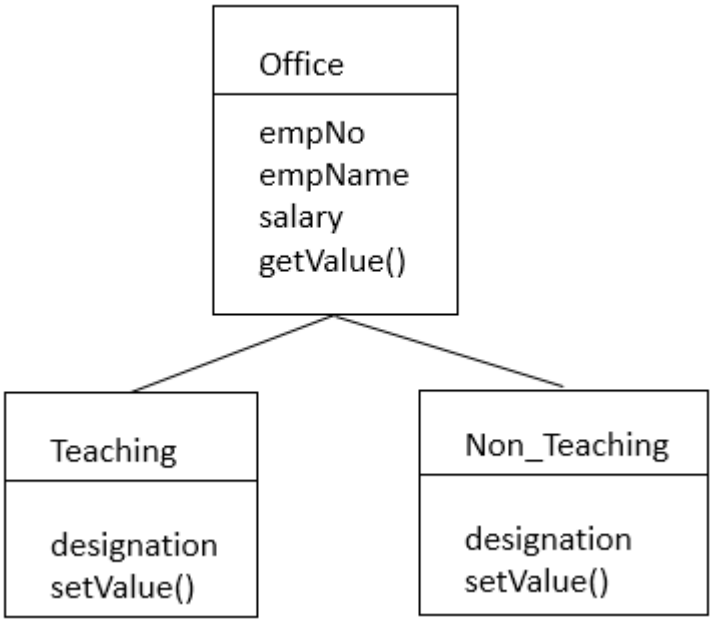
Subject Name: Java Programming **Semester:** III
Subject Code: CE241 **Academic year:** 2019-20

	PART-I Data Types, Variables, String, Control Statements, Operators, Arrays	LO	PO	PEO
1.	Introduction to Object Oriented Concepts, comparison of Java with other object oriented programming languages. Introduction to JDK, JRE, JVM, javadoc, command line argument. Introduction to Eclipse or Netbean IDE and Console Programming.	1	1,3, 4,5	1,7
2.	Given a string, return a string made of the first 2 chars (if present), however include first char only if it is 'o' and include the second only if it is 'z', so "ozymandias" yields "oz". startOz("ozymandias") → "oz" startOz("bzoo") → "z" startOz("oxx") → "o"	1	1,3, 4,5	7
3.	Given two non-negative int values, return true if they have the same last digit, such as with 27 and 57. Note that the % "mod" operator computes remainders, so 17 % 10 is 7. lastDigit(7, 17) → true lastDigit(6, 17) → false lastDigit(3, 113) → true	1	1,3, 4,5	7
4.	Given an array of ints, return true if the sequence of numbers 1, 2, 3 appears in the array somewhere. array123([1, 1, 2, 3, 1]) → true array123([1, 1, 2, 4, 1]) → false array123([1, 1, 2, 1, 2, 3]) → true	1	1,3, 4,5	7
5.	Given 2 strings, a and b, return the number of the positions where they contain the same length 2 substring. So "xxcaazz" and "xxbaaz" yields 3, since the "xx", "aa", and "az" substrings appear in the same place in both strings. stringMatch("xxcaazz", "xxbaaz") → 3 stringMatch("abc", "abc") → 2 stringMatch("abc", "axc") → 0	1	1,3, 4,5	7
6.	**Computing Body Mass Index** You can use nested if statements to write a program that interprets body mass index. Body Mass Index (BMI) is a measure of health based on height and weight. It can be calculated by taking your weight in kilograms and dividing it by the square of your height in meters. The interpretation of BMI for people 20 years or older is as	1	1,3, 4,5	7

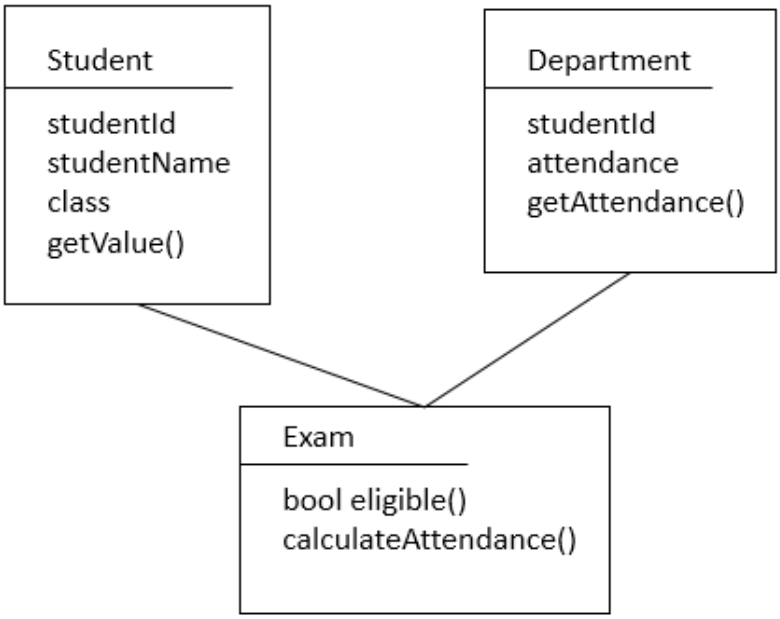
	follows: BMI Interpretation BMI < 18.5 Underweight 18.5 ≤ BMI < 25.0 Normal 25.0 ≤ BMI < 30.0 Overweight 30.0 ≤ BMI Obese Write a program that prompts the user to enter a weight in pounds and height in inches and displays the BMI. Note that one pound is 0.45359237 kilograms and one inch is 0.0254 meters. Listing 3.4 gives the program. Output- Enter weight in pounds: 146 Enter height in inches: 70 BMI is 20.948603801493316 Normal			
7.	**Lottery** The lottery program involves generating random numbers, comparing digits, and using Boolean operators. Suppose you want to develop a program to play lottery. The program randomly generates a lottery of a two-digit number, prompts the user to enter a two-digit number, and determines whether the user wins according to the following rules: 1. If the user input matches the lottery number in the exact order, the award is $10,000. 2. If all digits in the user input match all digits in the lottery number, the award is $3,000. 3. If one digit in the user input matches a digit in the lottery number, the award is $1,000. Note that the digits of a two-digit number may be 0. If a number is less than 10, we assume the number is preceded by a 0 to form a two-digit number. For example, number 8 is treated as 08 and number 0 is treated as 00 in the program. Listing 3.8 gives the complete program. **Output: Test Cases** Case-1 Enter your lottery pick (two digits): 15 The lottery number is 15 Exact match: you win $10,000 Case-2 Enter your lottery pick (two digits): 45 The lottery number is 54 Match all digits: you win $3,000 Case-3 Enter your lottery pick: 23 The lottery number is 34	1	1,3, 4,5	7

	Match one digit: you win $1,000 Case-4 Enter your lottery pick: 23 The lottery number is 14 Sorry: no match			
8.	The problem is to write a program that will grade multiple-choice tests. Assume there are eight students and ten questions, and the answers are stored in a two-dimensional array. Each row records a student's answers to the questions, as shown in the following array. Students' Answers to the Questions: 0 1 2 3 4 5 6 7 8 9 Student 0 A B A C C D E E A D Student 1 D B A B C A E E A D Student 2 E D D A C B E E A D Student 3 C B A E D C E E A D Student 4 A B D C C D E E A D Student 5 B B E C C D E E A D Student 6 B B A C C D E E A D Student 7 E B E C C D E E A D The key is stored in a one-dimensional array: Key to the Questions: 0 1 2 3 4 5 6 7 8 9 Key D B D C C D A E A D Your program grades the test and displays the result. It compares each student's answers with the key, counts the number of correct answers, and displays it.	1	1,3, 4,5	7
9.	The problem is to check whether a given Sudoku solution is correct.  (a) Puzzle (b) Solution	1	1,3, 4,5	7
10.	**Finding a Closest Pair** The GPS navigation system is becoming increasingly popular. The system uses the graph and geometric algorithms to calculate distances and map a route. The practical presents a geometric problem for finding a closest pair of point. **Output:** Enter the number of points: 8 Enter 8 points: (-1, 3) (-1, -1) (1, 1) (2, 0.5) (2, -1) (3, 3) (4, 2) (4, -0.5) The closest two points are (1.0, 1.0) and (2.0, 0.5)	1	1,3, 4,5	7

PART-II Object Oriented Programming : Classes, Methods, Inheritance				
1.	Design a class named Circle containing following attributes and behavior. • One double data field named radius. The default value is 1. • A no-argument constructor that creates a default circle. • A Single argument constructor that creates a Circle with the specified radius. • A method named getArea() that returns area of the Circle. • A method named getPerimeter() that returns perimeter of it.	1,3	1,3, 4,5 6	3
2.	Design a class named Account that contains: • A private int data field named id for the account (default 0). • A private double data field named balance for the account (default 500₹). • A private double data field named annualInterestRate that stores the current interest rate (default 7%). Assume all accounts have the same interest rate. • A private Date data field named dateCreated that stores the date when the account was created. • A no-arg constructor that creates a default account. • A constructor that creates an account with the specified id and initial balance. • The accessor and mutator methods for id, balance, and annualInterestRate. • The accessor method for dateCreated. • A method named getMonthlyInterestRate() that returns the monthly interest rate. • A method named getMonthlyInterest() that returns the monthly interest. • A method named withdraw that withdraws a specified amount from the account. • A method named deposit that deposits a specified amount to the account.	1,3	1,3, 4,5 6	3
3.	Use the Account class created as above to simulate an ATM machine. Create 10 accounts with id **AC001…..AC010** with initial balance 300₹. The system prompts the users to enter an id. If the id is entered incorrectly, ask the user to enter a correct id. Once an id is accepted, display menu with multiple choices. 1. Balance inquiry 2. Withdraw money [Maintain minimum balance 300₹] 3. Deposit money 4. Money Transfer 5. Create Account 6. Deactivate Account 7. Exit **Hint**: Use **ArrayList**, which is can shrink and expand with compared to Array.	1,3	1,3, 4,5 6	3
4.	Write a java program that implements educational hierarchy using inheritance.	1,3	1,3, 4,5 6	3

				
5.	Develop a Program that illustrate method overloading concept.	1,3	1,3, 4,5 6	3

PART-III **Package & Interface**				
1.	WAP that illustrate the use of interface reference. Interface Luminious Object has two method lightOn() and lightOff(). There is one class Solid extended by 2 classes Cube and Cone. There is one class LuminiousCone extends Cone and implements Luminoius Interface. LumminuiousCube extends Cube and implements Luminious Interface. Create a object of LuminiousCone and LuminousCube and use the concept of interface reference to invoke the methods of interface.	3	1,4,5 6	3
2.	WAP that illustrate the interface inheritance. Interface P is extended by P1 and P2 interfaces. Interface P12 extends both P1 and P2. Each interface declares one method and one constant. Create one class that implemetns P12. By using the object of the class invokes each of its method and displays constant.	3	1,4,5 6	3
3.	Create an abstract class Robot that has the concretre subclasses , RobotA, RobotB, RobotC. Class RobotA1 extends RobotA, RobotB1 extends RobotB and RobotC1 extends RobotC. There is interface Motion that declares 3 methods forward(), reverse() and stop(), implemented by RobotB and RobotC. Sound interface declare method beep() implemented by RobotA1, RobotB1 and RobotC1. Create an instance method of each class and invoke beep() and stop() method by all objects.	3	1,4,5 6	3

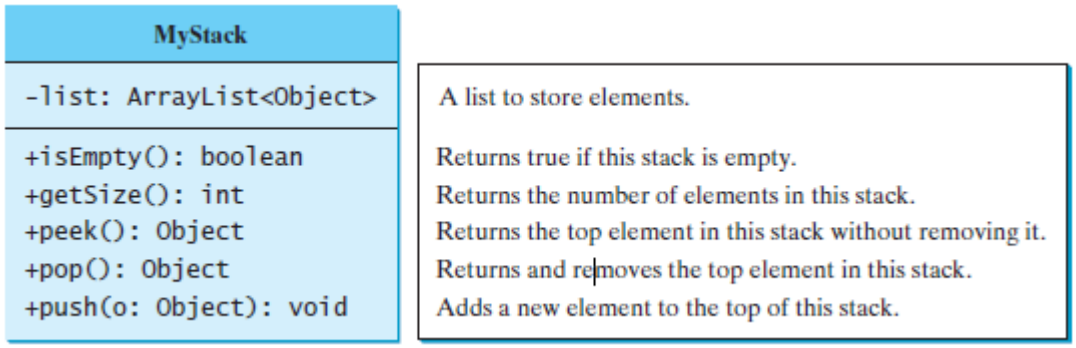
4.	Write a java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.	3	1,4,5 6	3
5.	Write a java program to find the details of the students eligible to enroll for the examination (Students, Department combinedly give the eligibility criteria for the enrollement class) using interfaces 	3	1,4,5 6	3
6.	Write a java program which shows importing of classes from other user define packages.	3	1,4,5 6	3
7.	Write a program that demonstrates use of packages & import statements.	3	1,4,5 6	3
8.	Write a program that illustrates the significance of interface default method.	3	1,4,5 6	3

PART-IV **Exception Handling**				
1.	WAP to show the try - catch block to catch the different types of exception.	3	1,4,5 6	3
2.	WAP to generate user defined exception using “throw” and “throws” keyword.	3	1,4,5 6	3
3.	Write a program that raises two exceptions. Specify two ‘catch’ clauses for the two exceptions. Each ‘catch’ block handles a different type of exception. For example the exception could be ‘ArithmeticException’ and ‘ArrayIndexOutOfBoundsException’. Display a message in the ‘finally’ block.	3	1,4,5 6	3

PART-V File Handling & Streams				
1.	WAP to show how to create a file with different mode and methods of File class to find path, directory etc.	2	2,3, 4,5	3,8
2.	Write a program to show a tree view of files and directories under a specified drive/volume.	2	2,3,	3,8
3.	Write a Java program that reads on file name from the user, then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes?	2	2,3,	3,8
4.	Write a program to transfer data from one file to another file so that if the destination file does not exist, it is created.	2	4,5	3,8
5.	WAP to show use of character and byte stream.	2	2,3,	3,8
6.	WAP to read console input and write them into a file. (BufferedReader /BufferedWriter).	2	4,5	3,8
7.	WAP to demonstrate methods of wrapper class.	2	2,3,	3,8

PART-VI Multithreading				
1.	Write a program to create thread which display "Hello World" message. A. by extending Thread class B. by using Runnable interface.	2	2,3, 4,5	3,8
2.	Write a program which takes N and number of threads as an argument. Program should distribute the task of summation of N numbers amongst number of threads and final result to be displayed on the console.	2	2,3, 4,5	3,8
3.	Write a java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.	2	2,3, 4,5	3,8
4.	Write a program to increment the value of one variable by one and display it after one second using thread using sleep() method.	2	2,3, 4,5	3,8
5.	Write a program to create three threads 'FIRST', 'SECOND', 'THIRD'. Set the priority of the 'FIRST' thread to 3, the 'SECOND' thread to 5(default) and the 'THIRD' thread to 7.	2	2,3, 4,5	3,8
6.	Write a program to solve producer-consumer problem using thread synchronization.	2	2,3, 4,5	3,8

PART-VII
Collection Framework and Generic

1.	Design a Custom Stack using ArrayList class, which implements following functionalties of stack. 			
2.	Create a generic method for sorting an array of Comparable objects.	2	2,3, 4,5	3,8
3.	Write a program that counts the occurrences of words in a text and displays the words and their occurrences in alphabetical order of the words. Using Map and Set Classes.	2	2,3, 4,5	3,8
4.	Write a code which counts the number of the keywords in a Java source file. Store all the keywords in a HashSet and use the contains method to test if a word is in the keyword set.	2	2,3, 4,5	3,8

PART-VII

1.	**Create and chat application using TCP/UDP protocol.** **(Note: Use Interface and Package concept)**	2	2,3, 4,5	3,8
2.	**Web Crawler** **This case study develops a program that travels the Web by following hyperlinks.** **The World Wide Web, abbreviated as WWW, W3, or Web, is a system of interlinked hypertext documents on the Internet. With a Web browser, you can view a document and follow the hyperlinks to view other documents. In this case study, we will develop a program that automatically traverses the documents on the Web by following the hyperlinks. This type of program is commonly known as a Web crawler. For simplicity, our program follows for the hyperlink that starts with http://.** **Figure shows an example of traversing the Web. We start from a Web page that contains three URLs named URL1, URL2, and URL3. Following URL1 leads to the page that contains three URLs named URL11, URL12, and URL13. Following URL2 leads to the page that contains two URLs named URL21 and URL22. Following URL3 leads to the page that contains four URLs named URL31, URL32, and URL33, and URL34. Continue to traverse the Web following the new hyperlinks. As you see, this process may continue forever, but we will exit the program once we have traversed 100 pages.**	2	2,3, 4,5	3,8

	The program follows the URLs to traverse the Web. To ensure that each URL is traversed only once, the program maintains two lists of URLs. One list stores the URLs pending for traversing and the other stores the URLs that have already been traversed. The algorithm for this program can be described as follows: **Add the starting URL to a list named listOfPendingURLs;** **while listOfPendingURLs is not empty and size of listOfTraversedURLs <= 100 {** **Remove a URL from listOfPendingURLs;** **if this URL is not in listOfTraversedURLs {** **Add it to listOfTraversedURLs;** **Display this URL;** **Read the page from this URL and for each URL contained in the page {** **Add it to listOfPendingURLs if it is not in listOfTraversedURLs;** **}** **}** **}** **Implement the above algorithm in programming.** **Output:** **Enter a URL: https://www.charusat.ac.in/** **Enter a URL: https://www.charusat.ac.in/all-courses/ug-programs/** **Craw https://www.charusat.ac.in/all-courses/ug-programs/** **Craw https://www.charusat.ac.in/all-courses/** **Craw https://www.charusat.ac.in/**			

Self Study: Control statements, flow control(Loops)

Beyond Syllabus: Lambda Expression, JJS, New features in Java9/10 (REPL, Functional Programming), JDBC,

Student Learning Outcome (LOs)

1. Students will be able to use different commands of JDK.
2. Student will able implement GUI as well as multithreaded programming for real life projects.
3. Students are able to design and develop projects in higher semesters using Object oriented design approach and java programming language.

The Programme Outcomes (POs)

4. Students with desirable knowledge, skills, positive attitude and professional behavior

5. Sharpening students aptitude, as required by the professional standards of excellence
6. Exposure of the graduates to the latest knowledge and skills, with practical hands-on experience
7. Industry-ready professional with a strong focus on delivering results according to the industry need and expectation
8. To enhance the employability and get right talent into the program
9. To revise course program to improve students' learning needs and curricula development, based on needs of society in general
10. Learning from international as well as domestic institutions and experts as they illustrate the best practices in their fields
11. Conductive to learning environment
12. To create and sustain the interest of the stakeholders in terms of Research, Quality Publications and Mobilization of the Resources
13. To increase and sustain the interest of the students and staff in professional society and its chapter related activities

The Programme Educational Objectives (PEOs)

1. To have up-to-date curricula of all the academic programs to meet the diverse and changing global industrial and societal needs, the various challenges and opportunities for the benefit of the students.
2. To have unique industry-oriented programs in collaboration with leading firms of the IT industry.

3. To develop and hone students' technical and behavioral competencies through appropriate pedagogical engagement regularly.

4. To foster student-faculty interaction programs for better understanding and appreciation of their mutual issues and evolving meaningful and appropriate mechanisms for effective learning by students.
5. To develop the professional competencies of faculty members and technical support staff.

6. To create an ambience where the students are cared for in every aspect and motivated to become excellent professionals who will continue to cherish their association with the faculty, staff and co-students.
7. To enhance Knowledge, skill and attitude.

8. To facilitate the development and evaluation of curricula**.**