


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

Subject Name: Java Programming
Subject Code: CE251

Semester: III
Academic year: 2019-20

PART-I		DATE	SIGN
Data Types, Variables, String, Control Statements, Operators, Arrays			
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.	24/6/19	
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"	26/6/19	
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	01/7/19	
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	01/7/19	
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	03/7/19	
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	03/7/19	

	<p>follows:</p> <table><tr><th>BMI</th><th>Interpretation</th></tr><tr><td>BMI < 18.5</td><td>Underweight</td></tr><tr><td>18.5 ≤ BMI < 25.0</td><td>Normal</td></tr><tr><td>25.0 ≤ BMI < 30.0</td><td>Overweight</td></tr><tr><td>30.0 ≤ BMI</td><td>Obese</td></tr></table> <p>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.</p> <p>Output-</p> <p>Enter weight in pounds: 146 Enter height in inches: 70 BMI is 20.948603801493316</p> <p>Normal</p>	BMI	Interpretation	BMI < 18.5	Underweight	18.5 ≤ BMI < 25.0	Normal	25.0 ≤ BMI < 30.0	Overweight	30.0 ≤ BMI	Obese		
BMI	Interpretation												
BMI < 18.5	Underweight												
18.5 ≤ BMI < 25.0	Normal												
25.0 ≤ BMI < 30.0	Overweight												
30.0 ≤ BMI	Obese												
7.	<p>Lottery</p> <p>The lottery program involves generating random numbers, comparing digits, and using Boolean operators.</p> <p>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:</p> <ol style="list-style-type: none">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. <p>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.</p> <p>Output: Test Cases</p> <p>Case-1 Enter your lottery pick (two digits): 15 The lottery number is 15 Exact match: you win \$10,000</p> <p>Case-2 Enter your lottery pick (two digits): 45 The lottery number is 54 Match all digits: you win \$3,000</p> <p>Case-3 Enter your lottery pick: 23 The lottery number is 34</p>	08/7/19											

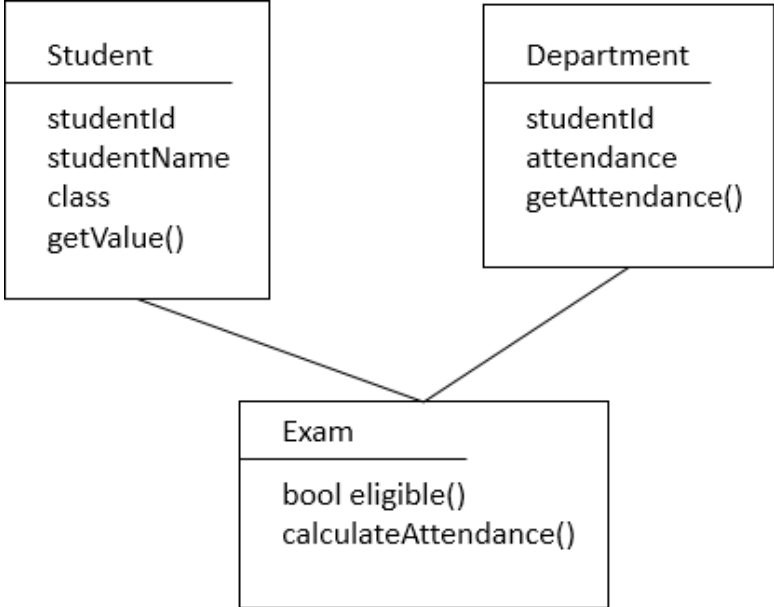
	Match one digit: you win \$1,000 Case-4 Enter your lottery pick: 23 The lottery number is 14 Sorry: no match																																																																																																																																																																																						
8.	<p>The 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.</p> <p>Students' Answers to the Questions:</p> <div>0 1 2 3 4 5 6 7 8 9 Student 0 A B A C C D E E A DStudent 1 D B A B C A E E A DStudent 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 DStudent 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</div> <p>The key is stored in a one-dimensional array:</p> <p>Key to the Questions:</p> <div>0 1 2 3 4 5 6 7 8 9 Key D B D C C D A E A D</div> <p>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.</p>	08/7/19																																																																																																																																																																																					
9.	<p>The problem is to check whether a given Sudoku solution is correct.</p> <div><table><tr><td>5</td><td>3</td><td></td><td></td><td>7</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>6</td><td></td><td></td><td>1</td><td>9</td><td>5</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>9</td><td>8</td><td></td><td></td><td></td><td></td><td></td><td>6</td><td></td></tr><tr><td>8</td><td></td><td></td><td></td><td>6</td><td></td><td></td><td></td><td></td><td>3</td></tr><tr><td>4</td><td></td><td></td><td>8</td><td></td><td>3</td><td></td><td></td><td></td><td>1</td></tr><tr><td>7</td><td></td><td></td><td></td><td>2</td><td></td><td></td><td></td><td></td><td>6</td></tr><tr><td></td><td>6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td>4</td><td>1</td><td>9</td><td></td><td></td><td></td><td>5</td></tr><tr><td></td><td></td><td></td><td></td><td>8</td><td></td><td></td><td></td><td>7</td><td>9</td></tr></table><p>(a) Puzzle</p></div> <div><p>Solution</p></div> <div><table><tr><td>5</td><td>3</td><td>4</td><td>6</td><td>7</td><td>8</td><td>9</td><td>1</td><td>2</td><td></td></tr><tr><td>6</td><td>7</td><td>2</td><td>1</td><td>9</td><td>5</td><td>3</td><td>4</td><td>8</td><td></td></tr><tr><td>1</td><td>9</td><td>8</td><td>3</td><td>4</td><td>2</td><td>5</td><td>6</td><td>7</td><td></td></tr><tr><td>8</td><td>5</td><td>9</td><td>7</td><td>6</td><td>1</td><td>4</td><td>2</td><td>3</td><td></td></tr><tr><td>4</td><td>2</td><td>6</td><td>8</td><td>5</td><td>3</td><td>7</td><td>9</td><td>1</td><td></td></tr><tr><td>7</td><td>1</td><td>3</td><td>9</td><td>2</td><td>4</td><td>8</td><td>5</td><td>6</td><td></td></tr><tr><td>9</td><td>6</td><td>1</td><td>5</td><td>3</td><td>7</td><td>2</td><td>8</td><td>4</td><td></td></tr><tr><td>2</td><td>8</td><td>7</td><td>4</td><td>1</td><td>9</td><td>6</td><td>3</td><td>5</td><td></td></tr><tr><td>3</td><td>4</td><td>5</td><td>2</td><td>8</td><td>6</td><td>1</td><td>7</td><td>9</td><td></td></tr></table><p>(b) Solution</p></div>	5	3			7						6			1	9	5						9	8						6		8				6					3	4			8		3				1	7				2					6		6												4	1	9				5					8				7	9	5	3	4	6	7	8	9	1	2		6	7	2	1	9	5	3	4	8		1	9	8	3	4	2	5	6	7		8	5	9	7	6	1	4	2	3		4	2	6	8	5	3	7	9	1		7	1	3	9	2	4	8	5	6		9	6	1	5	3	7	2	8	4		2	8	7	4	1	9	6	3	5		3	4	5	2	8	6	1	7	9		08/7/19	
5	3			7																																																																																																																																																																																			
6			1	9	5																																																																																																																																																																																		
	9	8						6																																																																																																																																																																															
8				6					3																																																																																																																																																																														
4			8		3				1																																																																																																																																																																														
7				2					6																																																																																																																																																																														
	6																																																																																																																																																																																						
			4	1	9				5																																																																																																																																																																														
				8				7	9																																																																																																																																																																														
5	3	4	6	7	8	9	1	2																																																																																																																																																																															
6	7	2	1	9	5	3	4	8																																																																																																																																																																															
1	9	8	3	4	2	5	6	7																																																																																																																																																																															
8	5	9	7	6	1	4	2	3																																																																																																																																																																															
4	2	6	8	5	3	7	9	1																																																																																																																																																																															
7	1	3	9	2	4	8	5	6																																																																																																																																																																															
9	6	1	5	3	7	2	8	4																																																																																																																																																																															
2	8	7	4	1	9	6	3	5																																																																																																																																																																															
3	4	5	2	8	6	1	7	9																																																																																																																																																																															
10.	<p>Finding a Closest Pair</p> <p>The GPS navigation system is becoming increasingly popular. The system uses the graph and geometric algorithms to calculate distances and map a route.</p> <p>The practical presents a geometric problem for finding a closest pair of point.</p> <p>Output:</p> <p>Enter the number of points: 8</p> <p>Enter 8 points:</p> <p>(-1, 3) (-1, -1) (1, 1) (2, 0.5) (2, -1) (3, 3) (4, 2) (4, -0.5)</p> <p>The closest two points are (1.0, 1.0) and (2.0, 0.5)</p>	10/7/19																																																																																																																																																																																					

	PART-II Object Oriented Programming : Classes, Methods, Inheritance	DATE	SIGN
1.	<p>Design a class named Circle containing following attributes and behavior.</p> <ul style="list-style-type: none"> • 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. 	17/7/19	
2.	<p>Design a class named Account that contains:</p> <ul style="list-style-type: none"> • 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. 	17/7/19	
3.	<p>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.</p> <ol style="list-style-type: none"> 1. Balance inquiry 2. Withdraw money [Maintain minimum balance 300₹] 3. Deposit money 4. Money Transfer 5. Create Account 6. Deactivate Account 7. Exit <p>Hint: Use ArrayList, which is can shrink and expand with compared to Array.</p>	17/7/19	
4.	Write a java program that implements educational hierarchy using inheritance.	17/7/19	

	<pre> classDiagram class Office { empNo empName salary getValue() } class Teaching { designation setValue() } class Non_Teaching { designation setValue() } Office < -- Teaching Office < -- Non_Teaching </pre>		
5.	Develop a Program that illustrate method overloading concept.	17/7/19	

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.	15/7/19	
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.	22/7/19	
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.	22/7/19	

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.	22/7/19	
5.	<p>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</p>  <pre> classDiagram class Student { studentId studentName class getValue() } class Department { studentId attendance getAttendance() } class Exam { eligible() calculateAttendance() } Student --> Exam Department --> Exam </pre>	24/7/19	
6.	Write a java program which shows importing of classes from other user define packages.	24/7/19	
7.	Write a program that demonstrates use of packages & import statements.	24/7/19	
8.	Write a program that illustrates the significance of interface default method.	29/7/19	

PART-IV

Exception Handling

1.	WAP to show the try - catch block to catch the different types of exception.	29/7/19	
2.	WAP to generate user defined exception using “throw” and “throws” keyword.	31/7/19	
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.	31/7/19	

	PART-V File Handling & Streams	DATE	SIGN
1.	WAP to show how to create a file with different mode and methods of File class to find path, directory etc.	07/8/19	
2.	Write a program to show a tree view of files and directories under a specified drive/volume.	07/8/19	
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?	07/8/19	
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.	07/8/19	
5.	WAP to show use of character and byte stream.	21/8/19	
6.	WAP to read console input and write them into a file. (BufferedReader /BufferedWriter).	21/8/19	
7.	WAP to demonstrate methods of wrapper class.	21/8/19	

	PART-VI Multithreading	DATE	SIGN
1.	Write a program to create thread which display “Hello World” message. A. by extending Thread class B. by using Runnable interface.	28/8/19	
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.	28/8/19	
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.	02/9/19	
4.	Write a program to increment the value of one variable by one and display it after one second using thread using sleep() method.	02/9/19	
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.	02/9/19	
6.	Write a program to solve producer-consumer problem using thread synchronization.	04/9/19	

	PART-VII Collection Framework and Generic	DATE	SIGN
1.	<p>Design a Custom Stack using ArrayList class, which implements following functionalities of stack.</p> <div> <div> MyStack - list: ArrayList<Object> +isEmpty(): boolean +getSize(): int +peek(): Object +pop(): Object +push(o: Object): void </div> <div> A list to store elements. Returns true if this stack is empty. Returns the number of elements in this stack. Returns the top element in this stack without removing it. Returns and removes the top element in this stack. Adds a new element to the top of this stack. </div> </div>	09/9/19	
2.	Create a generic method for sorting an array of Comparable objects.	09/9/19	
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.	11/9/19	
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.	11/9/19	

