JAVA PRACTISE QUESTION

1. Write the Java code for a class Calculator that has a main method. Inside main, declare two integer variables, num1 and num2, initialize them with values 10 and 5 respectively, and then print their sum. Ensure your class and method signatures are correctly structured.
2. Define a class PhysicsConstant and within it, declare a public static final double variable named GRAVITY initialized to 9.81. In the main method of this class, demonstrate how to access and print the value of GRAVITY. Pay attention to correct variable declaration and access.
3. Create a class CircleArea with a constructor that takes a double argument for the radius. Inside the constructor, store the radius in an instance variable. Additionally, include a method calculateArea that returns the area of the circle using Math.PI. Instantiate an object and call calculateArea in main.
4. Write a Java program for a class TemperatureConverter. Include a method celsiusToFahrenheit that takes a double Celsius temperature as input and returns the equivalent Fahrenheit temperature. In your main method, call this method with a sample Celsius value and print the result. Remember the formula: $F=C\times9/5+32$.
5. Design a class MathOperations with a method add that takes two integers and returns their sum. Overload this method with another add method that takes three integers and returns their sum. In main, demonstrate calling both overloaded add methods.

6. Implement a class Student with a constructor that takes String name and int age. Include a method displayStudentInfo that prints the student's name and age. In main, create two Student objects and call displayStudentInfo for each.
7. Develop a class DistanceCalculator that has a method calculateDistance which takes two double coordinates, x1 and x2, and returns the absolute difference between them. Use Math.abs() for this calculation. In your main method, test this method with different coordinate pairs.
8. Write a Java program to define a class GeometricShape. Within this class, define a method printType that takes a String shapeName as input and prints "This is a " followed by the shape name. In main, create an object of GeometricShape and call printType with "Square".
9. Create a class Counter with a private integer instance variable count. Include a public constructor that initializes count to 0. Add a method increment that increases count by 1 and a method getCount that returns the current count value. In main, create a Counter object, increment it a few times, and print the count.
10. Implement a class PowerCalculator that has a method calculatePower taking two integer arguments, base and exponent. This method should return base raised to the power of exponent using Math.pow(). In your main method, call calculatePower with 2 and 3 and print the result.
11. Design a class EquationSolver that has a main method. Inside main, declare an integer x and initialize it to 5. Use

an if statement to check if x is greater than 0. If it is, print "x is positive". Otherwise, print "x is not positive".

12. Write a Java program for a class EvenOddChecker. Include a method checkEvenOdd that takes an integer as input and prints whether it's "Even" or "Odd" using an if-else statement. In main, call this method with 7 and then with 10.

13. Create a class GradeEvaluator with a main method. Declare an integer variable score and set it to 85. Use an if-else if-else structure to print "Excellent" if score is 90 or above, "Good" if score is 70 or above, and "Needs Improvement" otherwise.

14. Define a class LoopExample with a main method. Use a for loop to print numbers from 1 to 5 (inclusive). Ensure the loop syntax is correct.

15. Implement a class FactorialCalculator that has a method calculateFactorial which takes an integer n and returns its factorial using a for loop. The factorial of 0 is 1. In main, call calculateFactorial with 4 and print the result.

16. Develop a class Summation that has a main method. Use a while loop to calculate the sum of numbers from 1 to 10. Print the final sum.

17. Write a Java program for a class ReverseCounter with a main method. Use a for loop to print numbers from 10 down to 1.

18. Create a class MultiplicationTable with a method printTable that takes an integer num and uses a for loop

to print the multiplication table for num up to 10. In main, call printTable for 7.

19. Design a class GuessingGame with a main method. Generate a random integer between 1 and 10 (you can hardcode 5 for now). Use a do-while loop to repeatedly ask the user (simulated by printing) to "Guess the number" until they guess the correct number.

20. Implement a class SwitchCaseExample with a main method. Declare a char variable grade and set it to 'B'. Use a switch statement to print "Good Job" for 'A', "Keep Going" for 'B', and "Try Harder" for 'C'. Include a default case for other grades.

21. Define a class WeekdayChecker with a method checkDay that takes an integer dayNum (1 for Monday, etc.). Use a switch statement to print the name of the weekday. In main, call checkDay with 3.

22. Create a class GeometricCalculator with a constructor that takes no arguments. Include a method calculateArea that returns 0.0. Overload this method to take double side and return the area of a square, and another overload to take double length, double width and return the area of a rectangle. Demonstrate all in main.

23. Write a Java program for a class VectorOperations. Include a method addVectors that takes two double arguments representing components of vector A (Ax,Ay) and two more double arguments for vector B (Bx,By). This method should print the resultant vector's components (Rx=Ax+Bx, Ry=Ay+By). In main, call this method with sample vector components.

24. Design a class PhysicsSimulation with a private double instance variable timeStep. Provide a constructor that initializes timeStep. Also, include a method simulateMotion that takes double initialVelocity and double acceleration and prints the final velocity after timeStep using Vf=Vi+a×timeStep.

25. Implement a class ArraySum with a main method. Declare an integer array numbers with elements {1, 2, 3, 4, 5}. Use a for-each loop to calculate and print the sum of all elements in the array.

26. Develop a class MaximumFinder that has a method findMax which takes two integers and returns the larger one. Overload this method to take three integers and return the largest among them. In main, test both findMax methods.

27. Write a Java program for a class SeriesSum. Include a method sumSeries that takes an integer n and calculates the sum of the first n natural numbers (1+2+⋯+n) using a for loop. In main, call this method with 5 and print the result.

28. Create a class Point with two private double instance variables x and y. Provide a constructor to initialize these. Add a method translate that takes double dx and double dy and updates x and y accordingly. In main, create a Point object, translate it, and print its new coordinates.

29. Define a class AverageCalculator with a method calculateAverage that takes an array of integers as input and returns their average as a double. Handle the case of

an empty array to avoid division by zero. In main, create an array and test this method.

30. Implement a class TriangleTypeChecker with a method checkTriangleType that takes three integer side lengths a, b, and c. Use if-else if-else to print "Equilateral" if all sides are equal, "Isosceles" if two sides are equal, and "Scalene" otherwise. In main, test with different side lengths.

31. Design a class NumberClassifier with a main method. Declare an integer num and set it to -3. Use nested if statements to first check if num is positive, negative, or zero. If positive, further check if it's even or odd. Print the appropriate classification.

32. Write a Java program for a class Countdown with a main method. Use a for loop with a break statement to print numbers from 10 down to 1, but stop if the number 7 is encountered.

33. Create a class SkipNumbers with a main method. Use a for loop with a continue statement to print numbers from 1 to 10, but skip 5 and 7.

34. Define a class UserAuthenticator with a constructor that takes String username and String password. Include a method authenticate that takes String inputUsername and String inputPassword and returns true if they match the stored credentials, false otherwise. In main, create an Authenticator object and test the authenticate method.

35. Implement a class SimpleQueue with a main method. Simulate adding elements to a queue using an ArrayList

and then removing them in a while loop until the queue is empty, printing each removed element.

36. Develop a class GeometricSeries with a method sumGeometricSeries that takes double a (first term), double r (common ratio), and int n (number of terms). Use a for loop to calculate and return the sum of the first n terms of a geometric series. In main, test with a=1,r=2,n=3.

37. Write a Java program for a class BankWithdrawal. Include a method withdraw that takes double amount and double currentBalance. Use an if statement to check if amount is less than or equal to currentBalance. If so, print "Withdrawal successful, new balance: " and the new balance; otherwise, print "Insufficient funds."

38. Create a class TrafficLight with a main method. Declare a String variable lightColor and set it to "Green". Use a switch statement to print "Go" for "Green", "Slow Down" for "Yellow", and "Stop" for "Red". Include a default case.

39. Define a class ArrayPrinter with a method printArray that takes an integer array and prints each element on a new line using a for loop. In main, create an array and call printArray.

40. Implement a class Rectangle with private double instance variables length and width. Provide a constructor to initialize these. Include methods getArea and getPerimeter. In main, create a Rectangle object and print its area and perimeter.

41. Design a class PhysicsEquation with a main method. Declare double mass = 10.0 and double acceleration = 2.0.

Calculate and print force = mass * acceleration. Ensure correct variable types and operations.

42. Write a Java program for a class VolumeCalculator. Include a method calculateVolume that takes double side and returns the volume of a cube (side3). Overload this method to take double radius, double height and return the volume of a cylinder (π×radius2×height). In main, demonstrate both.

43. Create a class Product with a constructor that takes String name and double price. Include a method applyDiscount that takes double percentage and returns the new price after applying the discount. In main, create a Product object, apply a discount, and print the discounted price.

44. Define a class LoopCondition with a main method. Use a while loop to print numbers from 1 to 5. Make sure the loop terminates correctly.

45. Implement a class GCDCalculator that has a method findGCD which takes two positive integers a and b. Use a while loop (Euclidean algorithm) to find and return their greatest common divisor. In main, test with 24 and 36.

46. Develop a class PrimeChecker with a method isPrime that takes an integer num and returns true if it's prime, false otherwise. Use a for loop for checking divisibility. In main, test with 11 and 12.

47. Write a Java program for a class SumEvenNumbers with a main method. Use a for loop to iterate from 1 to 10 and an if statement to add only even numbers to a sum variable. Print the total sum.

48. Create a class AreaComparison with a main method. Declare double radius = 5.0 and double squareSide = 8.0. Calculate the area of a circle and a square. Use an if-else statement to print which shape has a larger area.

49. Define a class UserMenu with a main method. Use a do-while loop to repeatedly display a menu (e.g., "1. Option A", "2. Option B", "3. Exit") and simulate user input until the user chooses to exit.

50. Implement a class StringManipulator with a method reverseString that takes a String and returns its reversed version using a for loop. In main, test with "hello".

51. Design a class Course with a constructor that takes String courseName and int creditHours. Include a method displayCourseInfo that prints the course name and credit hours. In main, create two Course objects and display their information.

52. Write a Java program for a class SpeedCalculator. Include a method calculateSpeed that takes double distance and double time and returns the speed. Handle the case where time is zero to prevent division by zero, returning 0.0 or a special value. In main, test with 100.0 distance and 10.0 time.

53. Create a class ArraySearch with a method findElement that takes an integer array and an integer target. Use a for loop to search for target in the array. If found, return its index; otherwise, return -1. In main, test this method.

54. Define a class TemperatureSensor with a private double instance variable currentTemperature. Provide a

constructor to initialize it. Add a method readTemperature that returns currentTemperature. In main, create a sensor, set its temperature, and read it.

55. Implement a class LoanCalculator with a method calculateMonthlyPayment that takes double principal, double annualInterestRate, and int years. Use a formula to calculate the approximate monthly payment. In main, test with sample values.

56. Develop a class NestedLoopExample with a main method. Use nested for loops to print a 3x3 grid of asterisks (*).

57. Write a Java program for a class EvenNumbersPrinter. Include a method printEven that takes an integer limit. Use a for loop and an if statement to print all even numbers from 1 up to limit. In main, call printEven with 15.

58. Create a class BankTransaction with a main method. Declare double balance = 500.0. Use a switch statement based on an int choice (e.g., 1 for deposit, 2 for withdrawal). Prompt (simulate) for an amount and update the balance accordingly.

59. Define a class StudentGrading with a constructor that takes int studentID and double midtermScore. Include a method calculateFinalGrade that takes double finalExamScore and returns the average of midterm and final exam scores. In main, create a StudentGrading object and calculate a final grade.

60. Implement a class ShapeArea with a static method calculateSquareArea that takes double side and returns

the area. Add another static method calculateRectangleArea that takes double length, double width and returns the area. In main, call these static methods directly using the class name.

61. Design a class UserInputValidator with a main method. Use a do-while loop to repeatedly ask the user (simulated) to "Enter a positive number" until a positive number is entered.

62. Write a Java program for a class PowerSeries with a method calculateExponent that takes double base and int exp. Use a for loop to calculate base raised to the power of exp without using Math.pow(). In main, test with 2 and 4.

63. Create a class VectorDotProduct with a method dotProduct that takes two integer arrays of equal length (representing vectors). Use a for loop to calculate and return their dot product. In main, test with {1, 2} and {3, 4}.

64. Define a class Book with private instance variables String title and String author. Provide a constructor to initialize them. Include a method displayBookInfo that prints the title and author. In main, create a Book object and display its information.

65. Implement a class TemperatureRanges with a main method. Declare double temperature = 25.0. Use nested if statements to check if the temperature is "Hot" (>30), "Warm" (20-30), or "Cold" (<20). If "Warm", further check if it's "Comfortable" (20-25) or "Mild" (25-30).

66. Develop a class FinancialPlanner with a method calculateFutureValue that takes double principal, double

annualRate, and int years. Use a for loop to calculate the future value with simple interest (FV=P(1+RT)). In main, test with sample values.

67. Write a Java program for a class VowelChecker. Include a method isVowel that takes a char and returns true if it's a vowel (a, e, i, o, u, case-insensitive), false otherwise, using a switch statement. In main, test with 'A' and 'b'.

68. Create a class FactorFinder with a method printFactors that takes an integer num. Use a for loop to print all factors of num. In main, call printFactors for 12.

69. Define a class LoginSystem with a main method. Use a while loop to allow up to 3 attempts for a simulated password (e.g., "password123"). If correct, print "Login successful"; otherwise, after 3 attempts, print "Account locked."

70. Implement a class StaticCounter with a static int count. Provide a static method incrementCount that increments count. Add a static method getCount that returns count. In main, call incrementCount multiple times and print getCount.

71. Design a class EquationRoot with a main method. Declare double a = 1, b = 5, c = 6. Calculate the discriminant delta = b*b - 4*a*c. Use if-else if-else to print "Two distinct real roots" if delta > 0, "One real root" if delta == 0, and "Complex roots" if delta < 0.

72. Write a Java program for a class ArrayManipulation. Include a method sumElements that takes an integer array and returns the sum. Overload this method to take

an integer array and a starting index and ending index, summing only elements within that range. Demonstrate both in main.

73. Create a class ProductDetails with a constructor that takes int productID, String productName, and double unitPrice. Include a method calculateTotalPrice that takes int quantity and returns the total price. In main, create a ProductDetails object and calculate a total price.

74. Define a class DoWhileLoopSum with a main method. Use a do-while loop to calculate the sum of numbers from 1 to 5. Ensure the loop structure is correct.

75. Implement a class CharacterClassifier with a main method. Declare a char variable ch = 'k'. Use if-else if-else to check if ch is an uppercase letter, a lowercase letter, a digit, or "Other character".

76. Develop a class MatrixPrinter with a method printMatrix that takes a 2D integer array (matrix) and prints its elements row by row. In main, create a sample 2x2 matrix and call printMatrix.

77. Write a Java program for a class GeometricMeanCalculator. Include a method calculateGeometricMean that takes two double numbers a and b and returns their geometric mean ($\sqrt{ab}$). Use Math.sqrt(). In main, test with 4.0 and 9.0.

78. Create a class Car with private String make and int year. Provide a constructor to initialize these. Include a method getAge that calculates and returns the car's age

assuming the current year is 2025. In main, create a Car object and print its age.

79. Define a class LoopControl with a main method. Use a for loop to print numbers from 1 to 10. If the number is 6, use break to exit the loop.

80. Implement a class SkipEvenNumbers with a main method. Use a for loop to print numbers from 1 to 10. If a number is even, use continue to skip printing it.

81. Design a class DataProcessor with a constructor that takes an integer array. Include a method processData that iterates through the array using a for-each loop and prints each element. In main, create a DataProcessor object and call processData.

82. Write a Java program for a class TimeConverter. Include a method secondsToMinutes that takes an integer seconds and returns the equivalent minutes and remaining seconds as a formatted string (e.g., "2 minutes and 30 seconds"). In main, test with 150 seconds.

83. Create a class UserValidation with a main method. Declare a String password = "secure" and String input = "wrong". Use a while loop to simulate asking for the password until it matches. Print "Access granted" or "Access denied" after 3 attempts.

84. Define a class StudentScores with a constructor that takes String studentName and int[] scores. Include a method calculateAverageScore that returns the average of the scores. In main, create a StudentScores object and calculate its average.

85. Implement a class ShapeDrawer with a method drawSquare that takes an integer size and uses nested for loops to print a square of asterisks (*) of that size. In main, call drawSquare with 4.

86. Develop a class ConditionalAssignment with a main method. Declare an integer value = 10. Use a ternary operator (? :) to assign "Even" to a String variable status if value is even, and "Odd" otherwise. Print status.

87. Write a Java program for a class TemperatureUnit. Include a method convert that takes double temp and String unit (e.g., "C" or "F"). Use a switch statement to convert the temperature to the other unit (Celsius to Fahrenheit or vice-versa) and return the converted value. In main, test with 25.0, "C".

88. Create a class CalculatorWithOverload with a method operate that takes two integers and returns their sum. Overload operate to take two double numbers and return their product. In main, test both operate methods.

89. Define a class NumberRangeChecker with a main method. Declare an integer number = 75. Use if-else if-else to check if number is in the range 0-50, 51-100, or above 100. Print the appropriate range.

90. Implement a class ArrayInitialization with a main method. Declare an integer array of size 5. Use a for loop to initialize each element with its index multiplied by 2. Then print the array elements.

91. Design a class SimplePhysics with a main method. Declare double initialVelocity = 5.0, double time = 2.0,

double acceleration = 1.5. Calculate and print the final velocity using Vf=Vi+at.

92. Write a Java program for a class StaticUtility. Include a static method maxOfTwo that takes two integers and returns the maximum. Add another static method minOfTwo that returns the minimum. In main, use these methods directly to find max and min of 8 and 3.

93. Create a class StudentGrade with a constructor that takes String studentName and char grade. Include a method isPassing that returns true if the grade is 'A', 'B', or 'C', false otherwise. In main, create a StudentGrade object and check if it's passing.

94. Define a class LoopSumLimit with a main method. Use a for loop to add numbers starting from 1 until the sum exceeds 20. Print the final sum and the last number added.

95. Implement a class NumberPattern with a main method. Use nested for loops to print the following pattern:

96. 1

97. 12

98. 123

99. 1234

100. 12345

101. Develop a class PhysicsCalculations with a constructor that takes double mass and double height. Include a method calculatePotentialEnergy that returns

the potential energy (PE=mgh, where g=9.81). In main, create an object and calculate its potential energy.

102. Write a Java program for a class DayOfWeekFinder. Include a method getDayName that takes an integer dayNum (1-7) and uses a switch statement to return the corresponding day name (e.g., "Monday", "Tuesday"). Handle invalid input. In main, test with 5.

103. Create a class ArrayStatistics with a method findMin that takes an integer array and returns the minimum element. Add another method findMax that returns the maximum element. In main, test with an array {3, 1, 4, 1, 5, 9, 2, 6}.

104. Define a class GeometricProperty with a constructor that takes double sideLength. Include a method calculatePerimeter that returns the perimeter of an equilateral triangle. Add a method calculateArea that returns the area of an equilateral triangle. In main, create an object and print both.

105. Implement a class LoopConditionalBreak with a main method. Use a while loop to print numbers from 1 up to 10. If the number is 4 and 6 then break from the loop.