

Exercise_04

1. Write java program using class which should need two functions : GetDetails() to get and PrintDetails() to print the details of a person. Details includes,
 - a. Name of the person.
 - b. Age of the person.
 - c. Phone number.
2. Write a RainFall class that stores the total rainfall for each of 12 months into an array of doubles. The program should have methods that return the following;
 - a. the total rainfall for the year
 - b. the average monthly rainfall
 - c. the month with the max rain
 - d. the month with the least rainDemonstrate the class in a complete program.
3. Write a Person class with an instance variable, *age* , and a constructor that takes an integer,*initialAge* , as a parameter. The constructor must assign *initialAge* to *age* after confirming the argument passed as *initialAge* is not negative; if a negative argument is passed as *initialAge* , the constructor should set age to 0 and print *Age is not valid, setting age to 0* and one more thing, constructor should call the amIOld() function to check the condition of the man.
amIOld() should perform the following conditional actions:
 - i. If *age*<13, print **you are YOUNG.**
 - ii. If *age*>13 and *age*<18, print **you are teenager.**
 - iii. Otherwise print **you are Old.**
4. Write a class 'Time' contains following things:
 - a. Three variables hours, minutes, seconds. Make sure that it should not be visible outside the class.
 - b. Define three functions
 - i. GetTime() : It read the time from user.
 - ii. AddTime(): To add the Time.
 - iii. DisplayTime(): to display the time.The time format should be HH:MM:SS.
Program should read two time from keyboard and find the sum of it and display it in proper order.
5. **Payroll Class**
Write a Payroll class that uses the following arrays as fields:
 - employeeId. An array of seven integers to hold employee identification numbers. The array should be initialized with the following numbers: 5658845 4520125 7895122 8777541 8451277 1302850 7580489
 - hours. An array of seven integers to hold the number of hours worked by each employee
 - payRate. An array of seven doubles to hold each employees hourly pay rate

- wages. An array of seven doubles to hold each employees gross wages

The class should relate the data in each array through the subscripts. For example, the number in element 0 of the hours array should be the number of hours worked by the employee whose identification number is stored in element 0 of the employeeId array. That same employee's pay rate should be stored in element 0 of the payRate array.

In addition to the appropriate accessor and mutator methods, the class should have a method that accepts an employees identification number as an argument and returns the gross pay for that employee. Demonstrate the class in a complete program that displays each employee number and asks the user to enter that employee's hours and pay rate. It should then display each employee's identification number and gross wages.

Hint: gross wage = (hourly pay) * (hours/week)*52/12.

6. Charge Account Validation

Create a class with a method that accepts a charge account number as its argument. The method should determine whether the number is valid by comparing it to the following list of valid charge account numbers:

5658845 4520125 7895122 8777541 8451277 1302850 8080152 4562555 5552012
5050552 7825877 1250255 L00523L 6545231 3852085 7576651 7881200 4581002

These numbers should be stored in an array or an **ArrayList** object. Use a sequential search to locate the number passed as an argument. If the number is in the array, the method should return true, indicating the number is valid. If the number is not in the array, the method should return false, indicating the number is invalid.

Problem Write a program that tests the class by asking the user to enter a charge account number. The program should display a message indicating whether the number is valid or invalid.

7. Driver's License Exam

The local Driver's License Office has asked you to write a program that grades the written portion of the driver's license exam. The exam has 20 multiple choice questions. Here are the correct answers:

1. B 6. A 11. B 16. C
2. D 7. B 12. C 17. C
3. A 8. A 13. D 18. B
4. A 9. C 14. A 19. D
5. C 10. D 15. D 20. A

A student must correctly answer 15 of the 20 questions to pass the exam. Write a class named **DriverExam** that holds the correct answers to the exam in an array field.

The class should also have an array field that holds the student's answers. The class should have the following methods:

- passed: returns true if the student passed the exam, or false if the student failed
- totalCorrect: returns the total number of correctly answered questions
- totalIncorrect: returns the total number of incorrectly answered questions
- questionsMissed: An int array containing the question numbers of the questions that the student missed

Demonstrate the class in a complete program that asks the user to enter a student's answers, and then displays the results returned from the **DriverExam** class's methods.

8. Grade Book

A teacher has five students who have taken four tests. The teacher uses the following grading scale to assign a letter grade to a student, based on the average of his or her four test scores:

Test Score	Letter Grade
90-100	A
80-89	B
70-79	C
60-69	D
0-39	F

Write a class that uses a string array or an ArrayList object to hold the five student's names, an array of five characters to hold the five students' letter grades, and five arrays of four doubles each to hold each student's set of test scores. The class should have methods that return a specific student's name, the average test score, and a letter grade based on the average. Demonstrate the class in a program that allows the user to enter each student's name and his or her four test scores. It should then display each student's average test score and letter grade.

9. Lottery Application

Write a Lottery class that simulates a lottery. The class should have an array of five integers named lotteryNumbers. The constructor should use the Random class (from the Java API) to generate a random number in the range of 0 through 9 for each element in the array. The class should also have a method that accepts an array of five integers that represent a person's lottery picks. The method is to compare the corresponding elements in the two arrays and return the number of digits that match. For example, the following shows the lotteryNumbers array and the user's array with sample numbers stored in each. There are two matching digits (elements 2 and 4).

Lottery Array:

7	4	9	1	3
---	---	---	---	---

User's Array:

4	2	9	7	3
---	---	---	---	---

In addition, the class should have a method that returns a copy of the lotteryNumbers array.

Demonstrate the class in a program that asks the user to enter five numbers. The program should display the number of digits that match the randomly generated lottery numbers. If all of the digits match, display a message proclaiming the user a grand prize winner.

10. Phone Book ArrayList

Write a class named PhoneBookEntry that has fields for a person's name and phone number. The class should have a constructor and appropriate accessor and mutator methods. Then write a program that creates at least five PhoneBookEntry objects and stores them in an ArrayList. Use a loop to display the contents of each object in the ArrayList.

11. A bank charges ₹25 per month plus the following check fees for a commercial checking Account:

₹25 each for fewer than 20 checks

₹20 each for 20-39 checks

₹15 each for 40-59 checks

₹10 each for 60 or more checks.

The bank also charges an extra ₹100 if the balance of the account falls Below ₹2000 (before any check fees are applied). Write a program that asks for the beginning balance and the number of checks written. Compute and display the bank service fees for the month.

12. Write a class named Coin. The Coin class should have the following field:

A String named sideUp. The sideUp field will hold either “heads” or “tails” indicating the side of the coin that is facing up.

The Coin class should have the following methods:

- A no-arg constructor that randomly determines the side of the coin that is facing up (“heads” or “tails”) and initializes the sideUp field accordingly.
- A void method named toss that simulates the tossing of the coin. When the toss method is called, it randomly determines the side of the coin that is facing up (“heads” or “tails”) and sets the sideUp field accordingly.
- A method named getSideUp that returns the value of the sideUp field.

Write a program that demonstrates the Coin class. The program should create an instance of the class and display the side that is initially facing up. Then, use a loop to toss the coin 20 times. Each time the coin is tossed, display the side that is facing up. The program should keep count of the number of times heads is facing up and the number of times tails is facing up, and display those values after the loop finishes.