



Sheet 3

Objective: upon successful completion of this sheet, students should be able to deal with arrays, jagged arrays, arrays of objects, and object exchanges.

1. Write Java programs, each contains one of the following methods.
 - The `insertAt()` method that takes as input the following: an array of integers, an integer element, and a position. Then the integer element is inserted in the array at the specified position.
 - The `findPrime()` method that takes an array of integers as input and returns another array contains only the prime numbers. Display the resulting array without using the loop statement.
 - The `getFactors()` method that takes an array of integers as input and returns a multi-dimensional array. The resulting array contains the factors of each element in the original array and whether each factor is even or odd. Even is represented by the value 1 and odd is represented by the value 0.
For example, if the given array contains the values {6,3}, then the resulting array will contain the following array of arrays for the first element, which is 6: {{1,0}, {2,1}, {3,0}, {6,1}}. For the second element, which is 3, it will look like this {{1,0}, {3,0}}.
2. Modify the question number 3 in the previous sheet, where each book can have multiple authors by replacing the instance variable of author by an array of authors. Modify the methods accordingly.

3. Design the “Rectangle” class that contains the following data members and methods.

Data members: length and width.

Methods: suitable constructor(s), setters, getters, `getPerimeter()`, and `getArea()`.

Test the class by creating an array of rectangles and displaying all the information, including the perimeter and the area of rectangles whose area exceeds 1500, using the `toString()` method.

Exchange rectangle objects in the defined array that have the same length but different width using the wrapper class way. Display all information about the rectangle objects before and after the exchange without using the loop statement.

4. Design the “Employee” class that contains the following data members and methods.

Data members: name, department, number of working hours per day, and salary.

Methods: suitable constructor(s), setters, getters, `addSalary()` which adds 500 EGP to the salary if it is less than 1000 EGP, and `addWork()` which adds 100 EGP to the salary if the working hours are more than 8.

Test the class by creating an array of employees and display all information about employees whose salary is less than 500.

Display also all information about employees whose working hours exceed 8 hours. Update the salaries of the employees according to the specified rules.

You should override the `toString()` method for displaying information about employees.