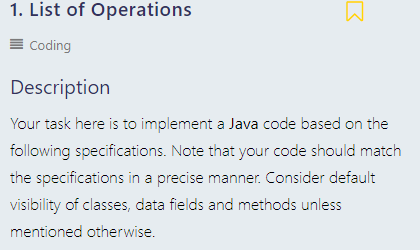
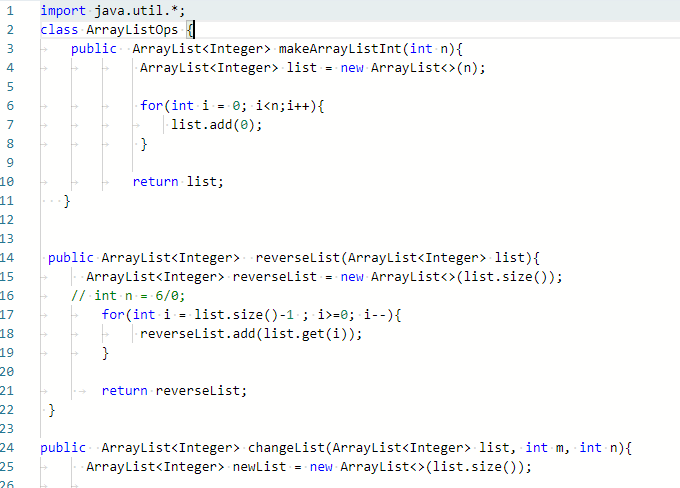
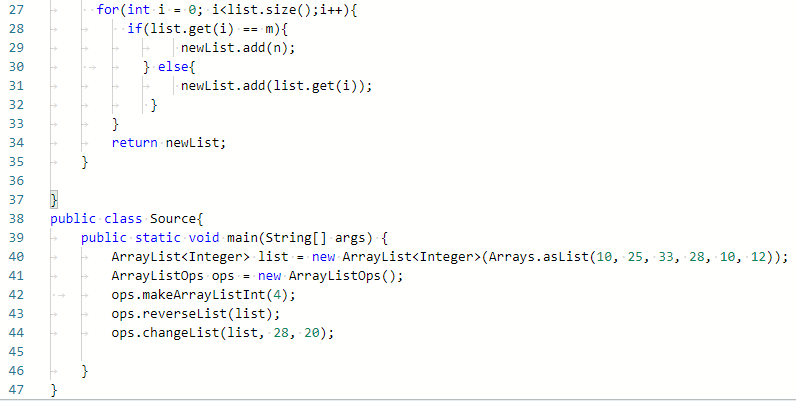
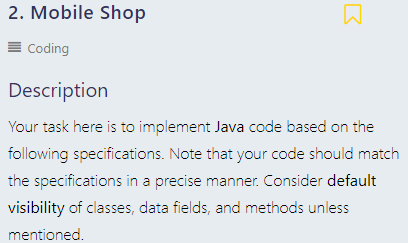
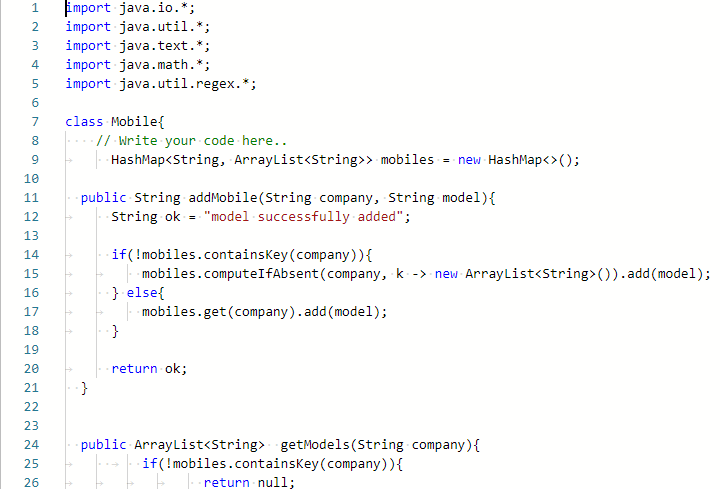
DoSelect Practice Test Solutions

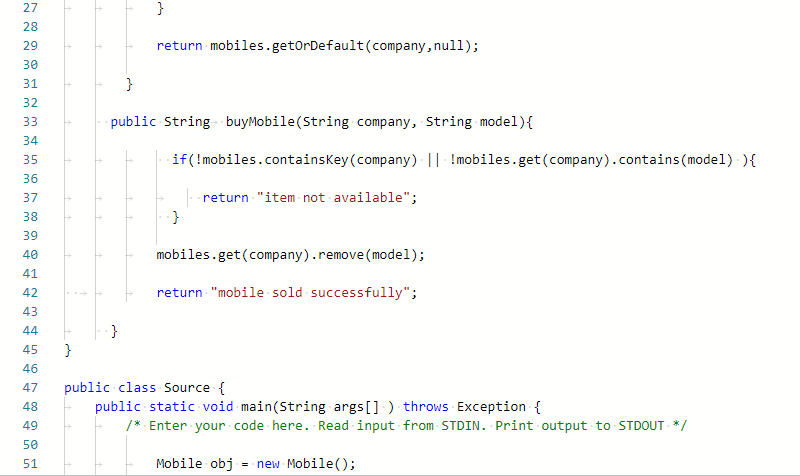


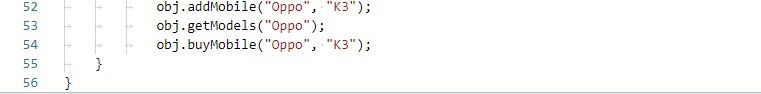


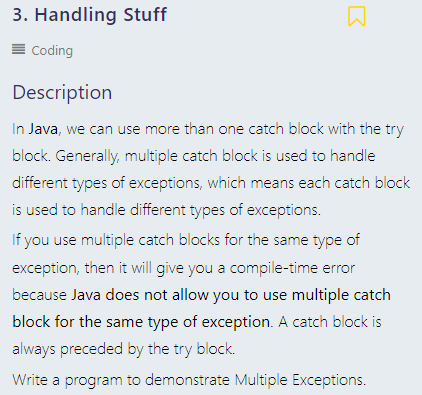


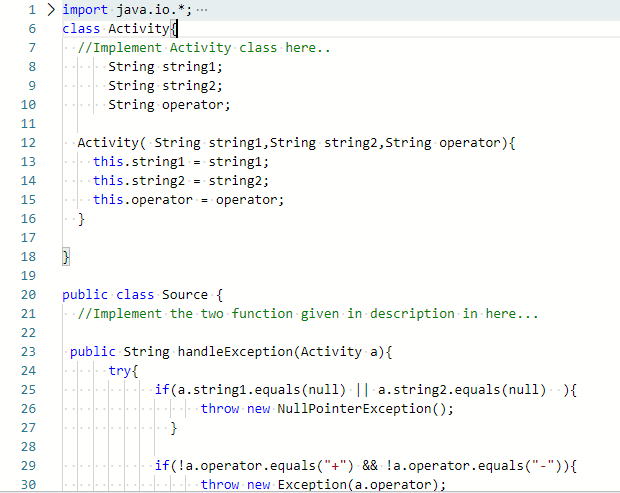


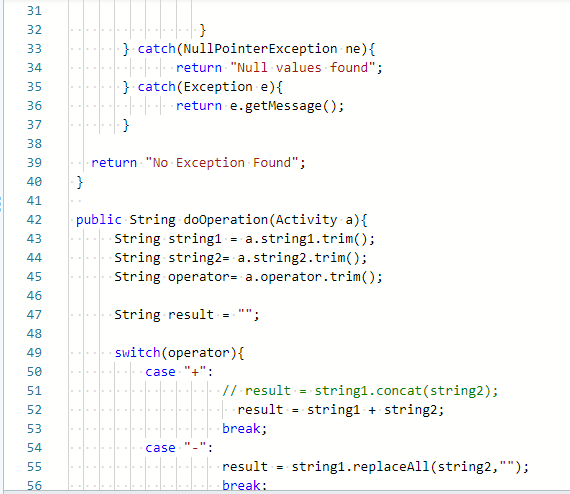


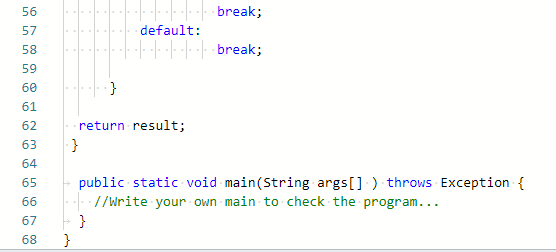


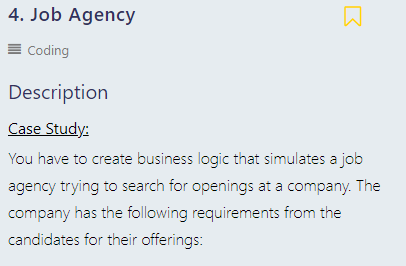


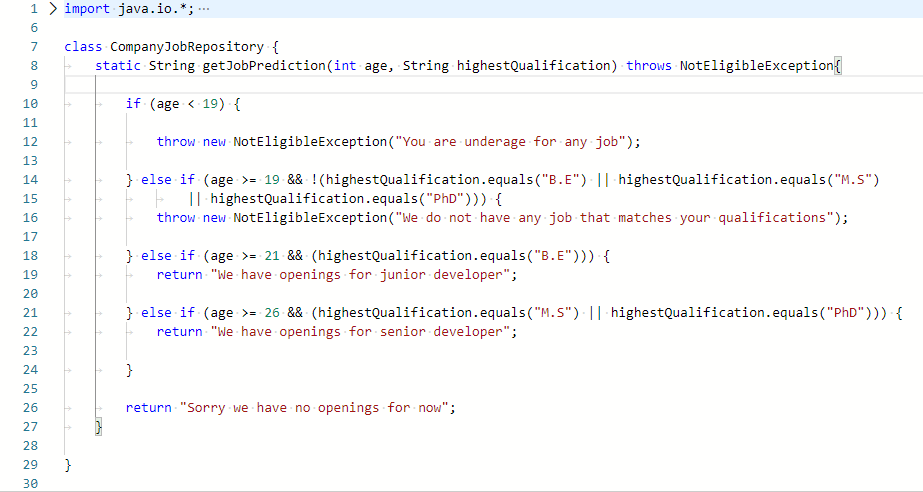


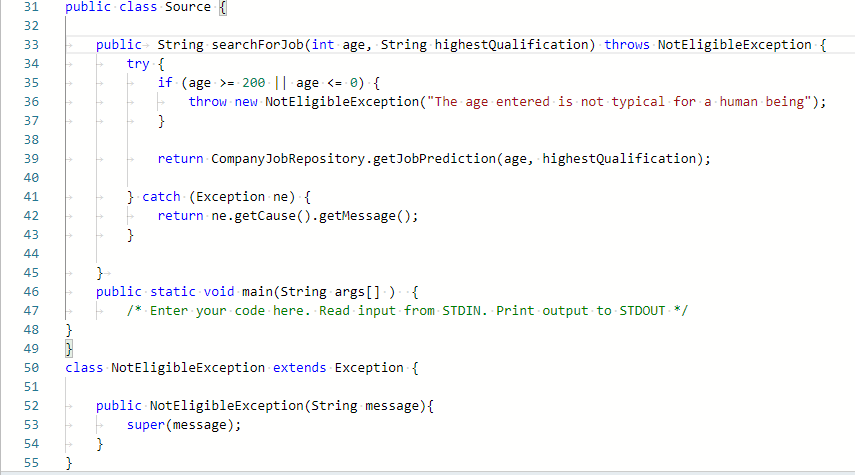


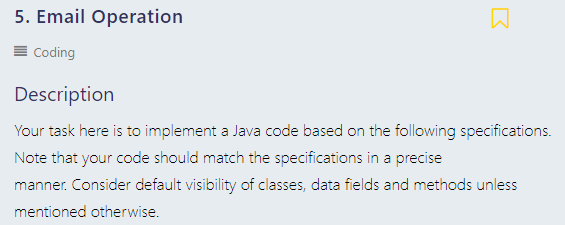


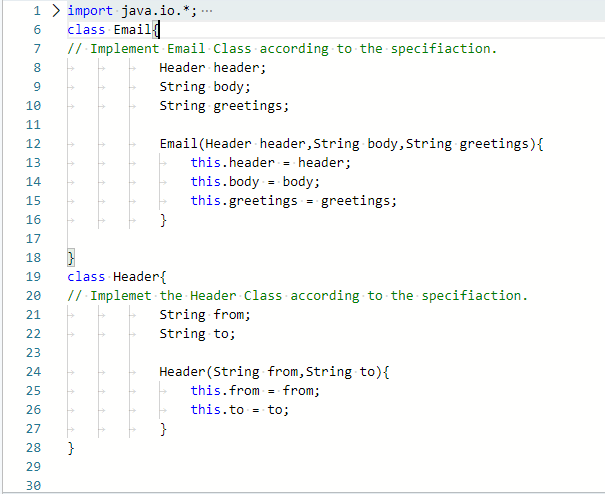


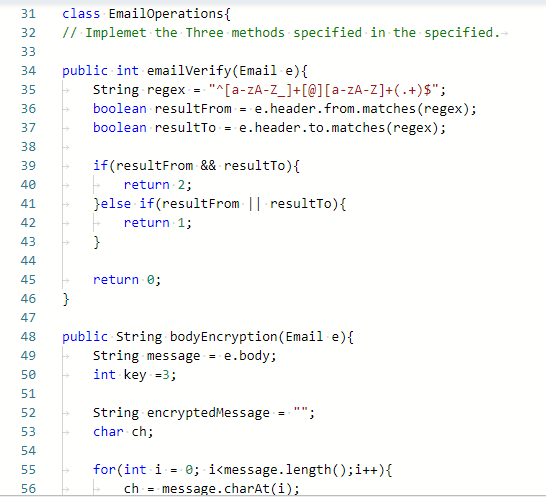


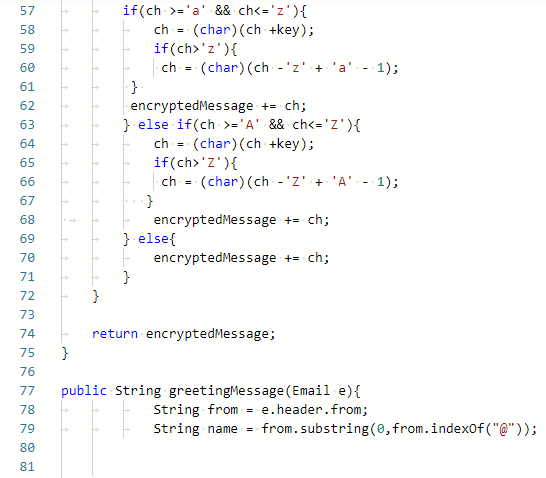


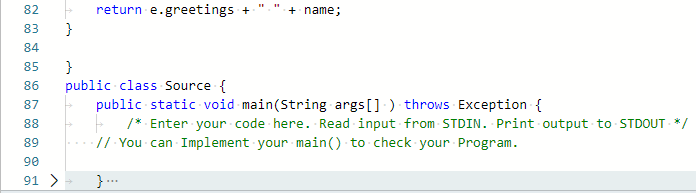


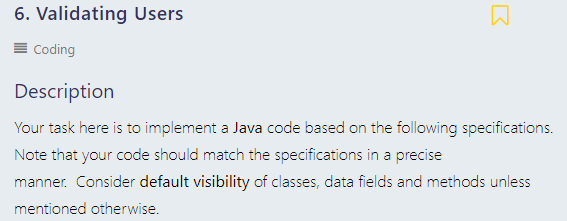


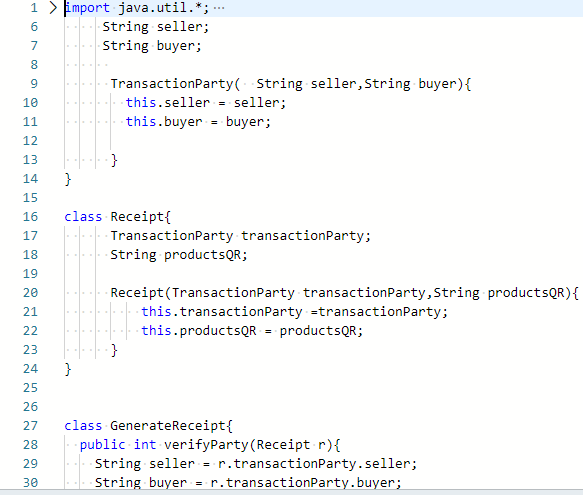


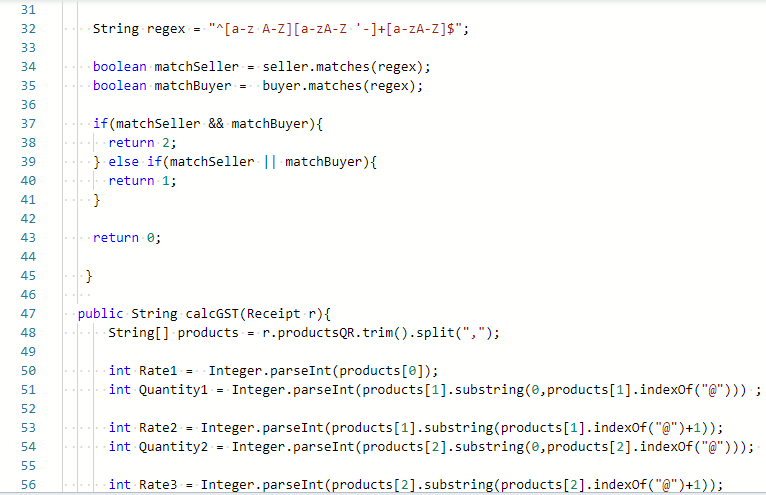


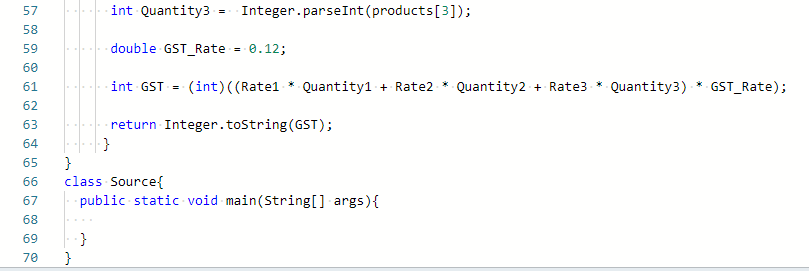


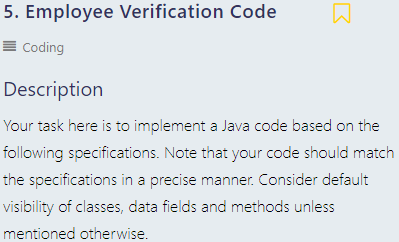


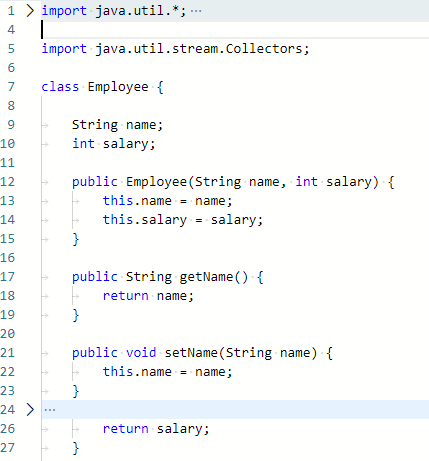


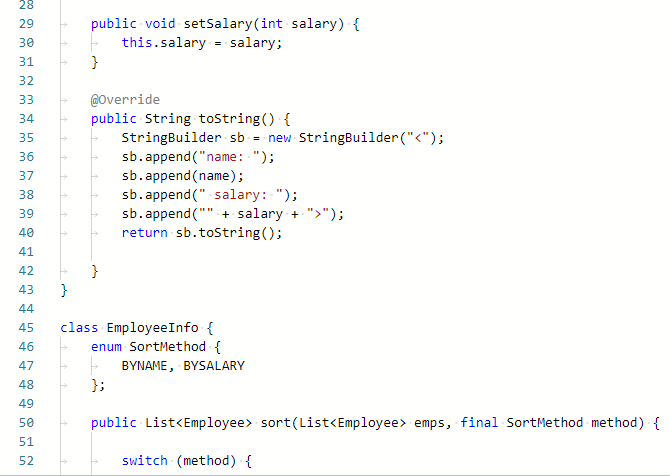


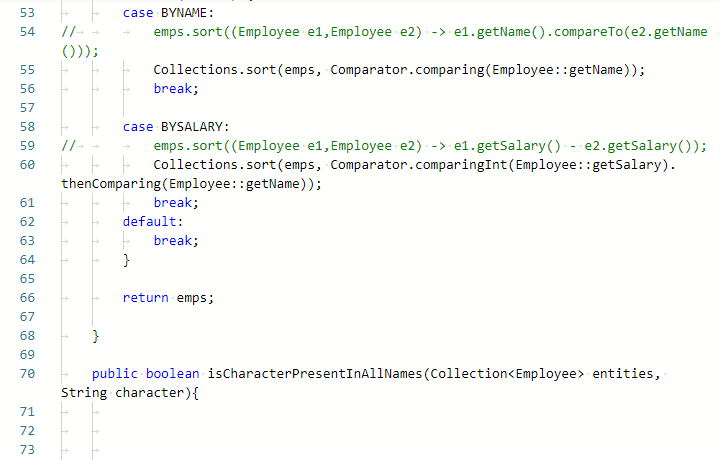


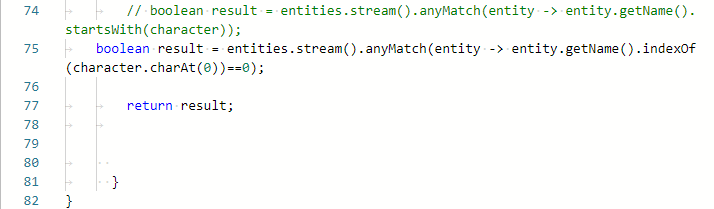


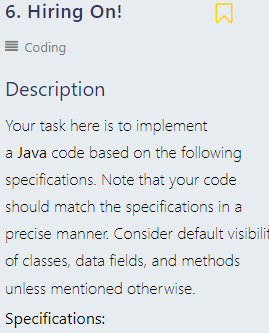


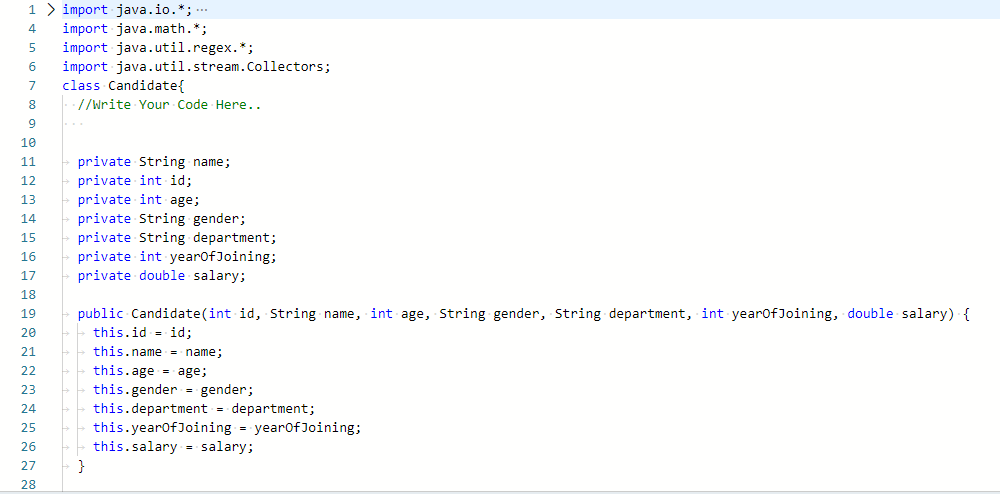


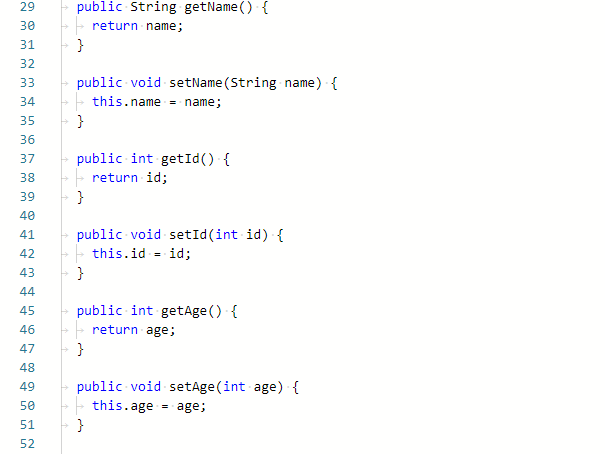


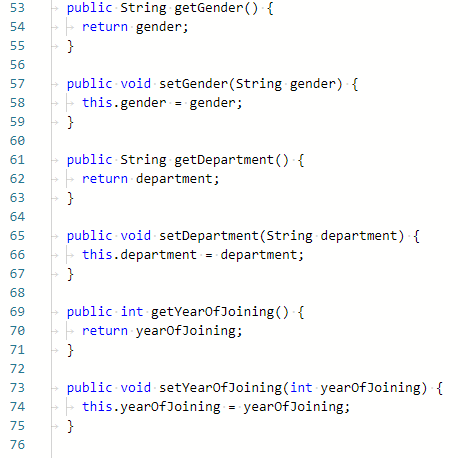


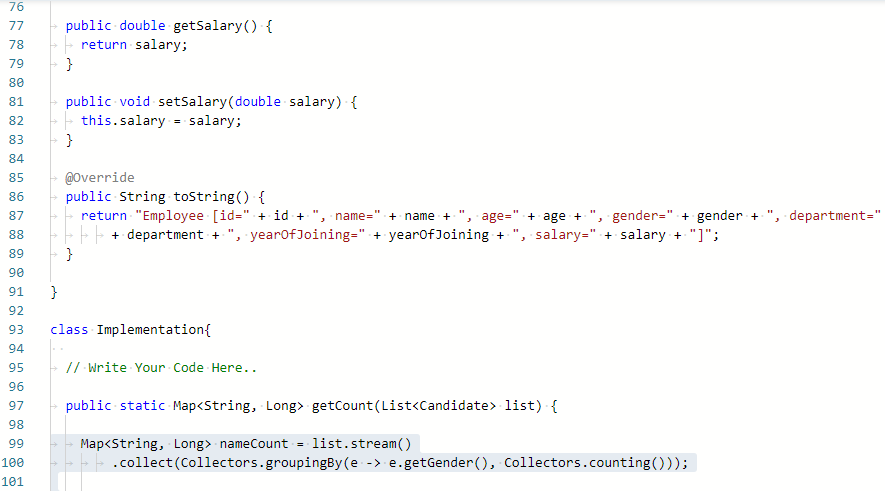


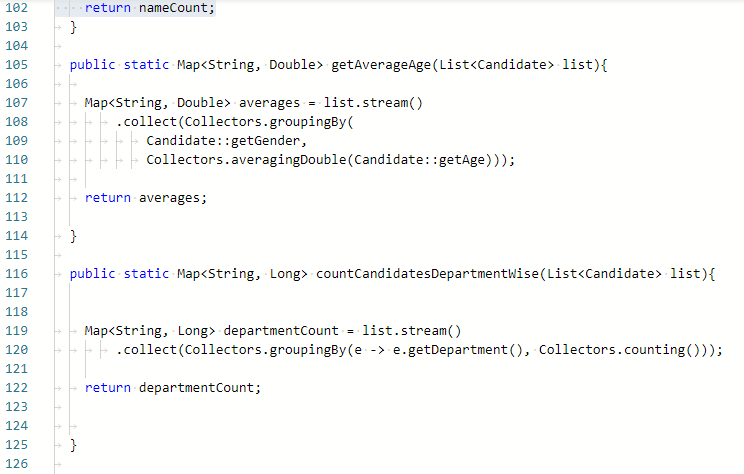


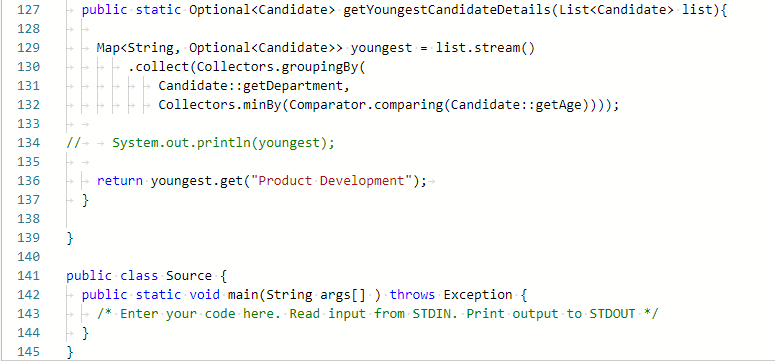












+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Sid recently started working for an online bookstore and has been given the task of creating a status maintenance system for all the books. Deadline is knocking at the door, and he is yet to finish his task. Can you help Sid develop the system?

He has created the base class Book as described below:

class Book:

data fields:

title : String variable which stores the title of the

book

author : String variable which stores the name of the author

(comma separated names if there are more than one author)

pages : Integer variable which stores the number of pages of the book

methods:

Book(String title, String author, int pages) :

Constructor to initialise the primary details

(title, author, pages) of the book

getDetails() :

Returns the title and author's name as a single

string separated by '-'

Now, your task is to define the following classes which are subclasses of Book class. Consider default visibility of classes, data fields and methods unless mentioned otherwise.

class Ebook:

data fields:

appxSize: int variable denoting the approximate size

of the book in kilo-bytes(KB)

format: String variable denoting the format of the e-book (mobi, pdf or epub)

methods:

Ebook(String title, String author, int pages, String format):

Constructor to initialise the details of the book and

calculate appxSize by calling setAppxSize()

setAppxSize():

Sets the approximate size of the Ebook calculated as:

15 times the number of pages for mobi format

10 times the number of pages for pdf format

5 times the number of pages for epub format

class PrintedBook:

data fields:

available: integer variable denoting the number of

available copies of the book // max value = 1000

basePrice: double variable denoting the base price of the book

sellingPrice : double variable denoting the final selling price of the book

methods:

PrintedBook(String title, String author, int pages, int available, double basePrice):

Constructor to initialise the details of the book

and calculate sellingPrice by calling setSellingPrice()

setSellingPrice():

Sets the selling price of the book calculated as:

add 5% tax and 2% of the quantity (1000-available)

to the base price to get the selling price

buy():

Decrease availability by 1 if it is greater than zero.

Update the selling price.You don't need to write the main function.

Input

(You don't need to process any input)

Output

(You don't need to output anything)

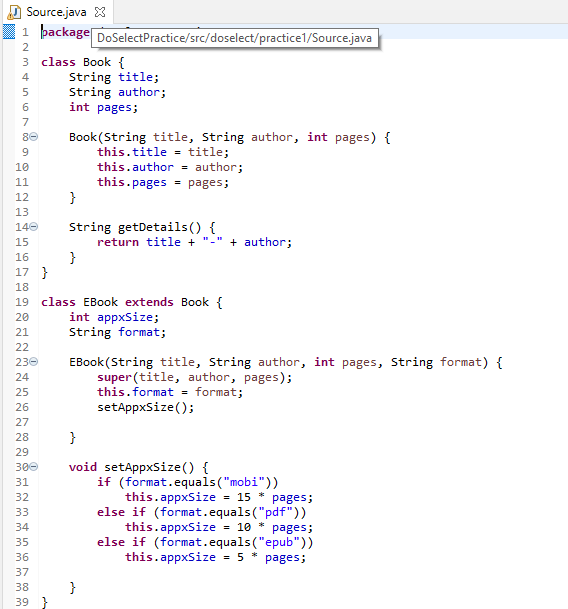
EXECUTION TIME LIMIT

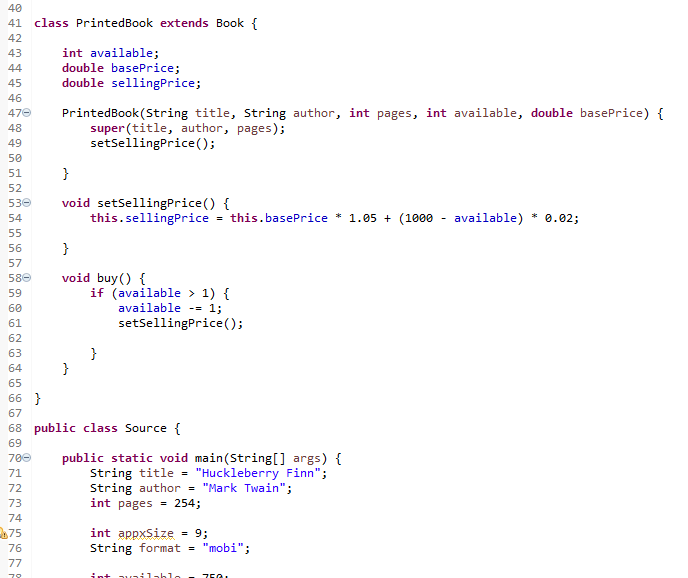
10 seconds

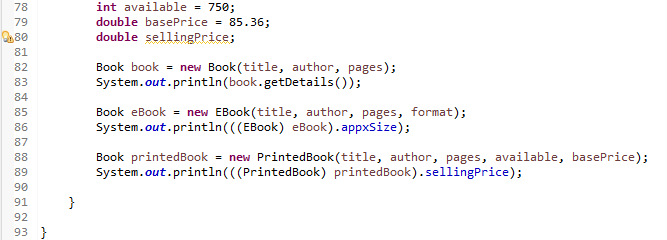
REPORT AN ISSUE

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Solution:**







\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Jake's school teacher gave him an assignment to write a Java program which calculates the area of a convex quadrilateral. The quadrilateral is described by the co-ordinates of four 2-dimensional points: (x1, y1), (x2, y2), (x3, y3) and (x4, y4). Jake is busy doing his Maths assignment. So, he asks for your help to complete the Java assignment for him, since you are good at it. He remembers that he has already done a similar assignment for triangle. So, he suggests you to take help of it.

The classes he had created already are described below. Consider default visibility of classes, data fields and methods unless mentioned otherwise.

class Point:

data fields:

x: integer variable denoting the x co-ordinate of the point

y: integer variable denoting the y co-ordinate of the point

methods:

Point:

Constructor to initialise the point

class Triangle:

data fields:

p1, p2, p3: Three Point objects denoting the

points that describes the triangle.

methods:

Triangle:

Constructor to initialise the triangle

getArea:

Returns a double variable denoting the area

of the triangle

Your task is to create a class named Quadrilateral which should be a subclass of Triangle. The description is given below:

class Quadrilateral extending the class Triangle:

data fields:

p4: The fourth point of the quadrilateral

methods:

Quadrilateral:

Constructor to initialize the quadrilateral

getArea: //overridden method

Returns a double variable denoting the area of

the quadrilateral, use the getArea method of

Triangle class to calculate it

Constraints

All the co-ordinates lie between -100 and 100.The points are given in either clockwise or anti-clockwise order.

You don't need to write the main function.

Input

An object of Triangle class or Quadrilateral class.

(You don't need to process any input)

Output

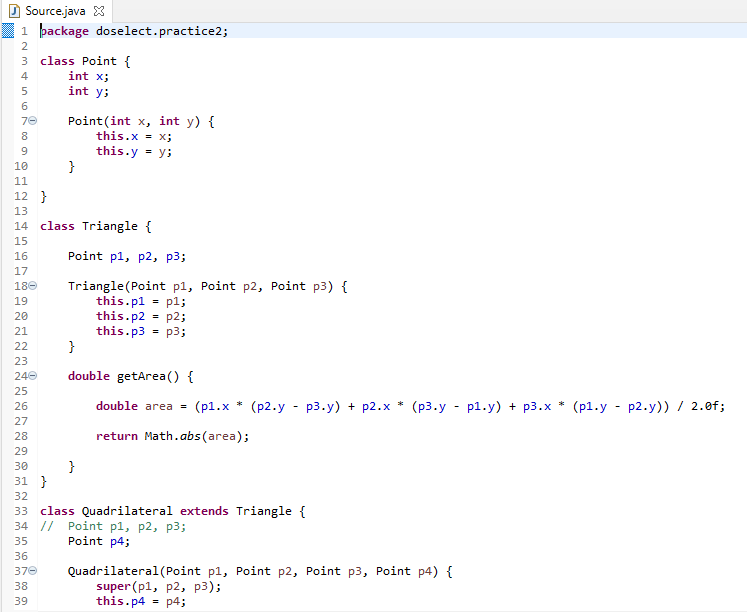
A double variable denoting the area of the Triangle or

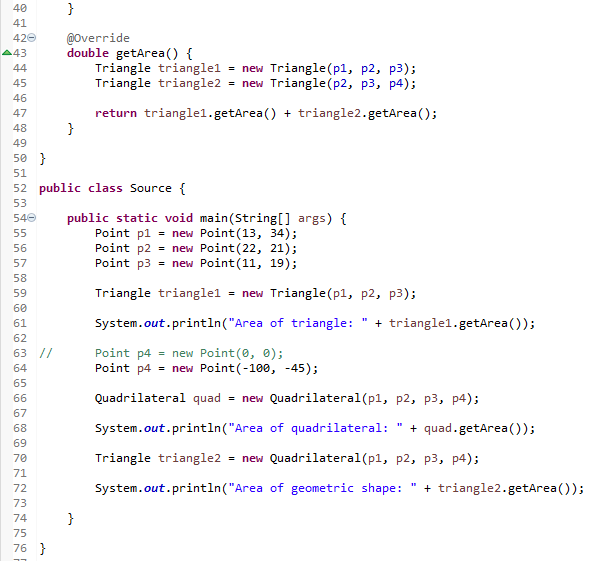
the Quadrilateral. (You don't need to output anything)

EXECUTION TIME LIMIT

10 seconds

**Solution:**





\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*