1. **Profit and Loss**

Sam purchased x dozens of toys at the rate of Rs. y per dozen. He sold each one of them at the rate of Rs. z. Can you please help him out with the percentage of profit?

Given the values of x, y and z, write a program to compute Sam's profit percentage.

**Example:**

Dozens of toys purchased (x) = 20

Rate per dozen (y) = Rs. 375

Selling price per toy (z) = Rs. 33

Cost Price of 1 toy = 375/12 = Rs. 31.25

Selling Price of 1 toy = Rs.33

Profit = 33 - 31.25 = Rs. 1.75

Profit % = 1.75 / 31.25 \* 100 = 5.6%

**Input Format:**

Input consists of 3 integers – x, y and z.

x - Number of dozens purchased.

y - Cost per dozen.

z - Selling price per item.

**Output Format:**

Refer Sample Input and Output for formatting details. The profit percentage needs to be printed correct to 2 decimal places.

**Sample Input and Output:**

Enter the number of dozens of toys purchased

**20**

Enter the price per dozen

**375**

Enter the selling price of 1 toy

**33**

Sam's profit percentage is 5.60 percent

1. **Discount Calculation**

Calculate the discount based on the price of two items and the overall discount percentage.

**Input Format:**

1. Item 1 price as floating point
2. Item 2 price as floating point
3. Discount as integer

**Output Format:**

1. Total of Item 1 and Item 2
2. Price after discount (correct to 2 decimal places)
3. Amount discounted (correct to 2 decimal places)

**Sample Input and Output:**

Price of item 1 :

**20.50**

Price of item 2 :

**45.40**

Discount in percentage :

**10**

Total amount : $65.90

Discounted amount : $59.31

Saved amount : $6.59

3) **Compare 2 Integers**

Write a program to relate 2 integers entered by the user as equal to, less than or greater than.

**Input and Output Format:**

Input consists of 2 integers.

**Sample Input and Output 1:**

Enter the first number

**6**

Enter the second number

**8**

6 is less than 8

**Sample Input and Output 2:**

Enter the first number

**8**

Enter the second number

**6**

8 is greater than 6

**Sample Input and Output 3:**

Enter the first number

8

Enter the second number

8

8 is equal to 8

4) Secure URL

  Write a program to check whether the given URL is secure.

**Example:**

Secure URL: https://www.amazon.com/

**Sample Input and Output 1:**

Enter the string

[**http://www.amazon.com/**](http://www.amazon.com/)

Enter the start string

**https**

"[http://www.amazon.com/](https://www.amazon.com/)" does not start with "https"

**Sample Input and Output 2:**

Enter the string

[**https://www.amazon.com/**](https://www.amazon.com/)

Enter the start string

**https**

"<https://www.amazon.com/>" starts with "https"

**5) Replace Organization Name**

This program is to illustrate the use of the method replace() defined in the string API.

Two companies enter into a Marketing Agreement and they prepare an Agreement Draft. After that one of the companies changes its name. The name changes need to be made in the Agreement Draft as well. Write a program to perform the name changes in the agreement draft.

**Sample Input and Output :**

Enter the content of the document

**ITT is a private organisation. ITT is a product based company. DBox is a ITT product**

Enter the old name of the company

**ITT**

Enter the new name of the company

**TTT**

The content of the modified document is

**TTT is a private organisation. TTT is a product based company. DBox is a TTT product**

6) **Profit and Loss (Using Methods)**

Sam purchased x dozens of toys at the rate of Rs. y per dozen. He sold each one of them at the rate of Rs. z. Can you please help him out percentage of profit?

Given the values of x, y and z, write a program to compute Sam's profit percentage.

**Example:**

Dozens of toys purchased (x) = 20

Rate per dozen (y) = Rs. 375

Selling price per toy (z) = Rs. 33

Cost Price of 1 toy = 375/12 = Rs. 31.25

Selling Price of 1 toy = Rs.33

Profit = 33 - 31.25 = Rs. 1.75

Profit % = 1.75 / 31.25 \* 100 = 5.6%

**Note:**

* Use methods to modularize the program coded earlier for this problem statement.
* Create a method calculateProfit() with the below mentioned signature.

public float calculateProfit(int dozenCount, int pricePerDozen, int sellPrice)

* Invoke this method from the main method.

I**nput Format:**

Input consists of 3 integers – x, y and z.

x - Number of dozens purchased.

y - Cost per dozen.

z - Selling price per item.

**Output Format:**

Refer Sample Input and Output for formatting details. The profit percentage needs to be printed correct to 2 decimal places.

**Sample Input and Output:**

Enter the number of dozens of toys purchased

20

Enter the price per dozen

375

Enter the selling price of 1 toy

33

Sam's profit percentage is 5.60 percent

7)

**Print Multiplication Table (30 minutes)**   
  
For any given number print the multiplication table.  
  
**Sample Input:**  
5  
**Sample Output:**  
5 x 1 = 5  
5 x 2 = 10  
5 x 3 = 15  
5 x 4 = 20  
5 x 5 = 25  
5 x 6 = 30  
5 x 7 = 35  
5 x 8 = 40  
5 x 9 = 45  
5 x 10 = 50

8)

**Set of Boxes**

 A manufacturing company have received multiple Boxes of raw materials. Write a program to store the box details into a Set.

**Problem Constraint:**

1. Create a class named Box with attributes length, width and height. All attributes should be of type double. Implement 3 argument constructor, setter / getters and toString() method.
2. Create a Set to store details of multiple Boxes.
3. The Set should have Boxes with unique volume.
4. When adding a Box into the Set, if there is a Box already present with the same volume in the Set, then it should not be added to the Set. Override equals() method in Box to achieve this functionality.

**Sample Input and Output :**  
Enter the number of Box   
**5**  
Enter the Box 1 details   
Enter Length   
**2.1**  
Enter Width   
**1.2**   
Enter Height   
**2.1**  
Enter the Box 2 details   
Enter Length   
**3.2**  
Enter Width   
**2.3**   
Enter Height   
**3.2**  
Enter the Box 3 details   
Enter Length   
**1.2**   
Enter Width   
**2.1**  
Enter Height   
**1.2**  
Enter the Box 4 details   
Enter Length   
**3.2**  
Enter Width   
**2.3**  
Enter Height   
**3.2**   
Enter the Box 5 details   
Enter Length   
**3.3**   
Enter Width   
**2.2**  
Enter Height   
**1.1**   
Unique Boxes in the Set are   
Length =1.2 Width =2.1 Height =1.2 Volume =3.02   
Length =2.1 Width =1.2 Height =2.1 Volume =5.29   
Length =3.3 Width =2.2 Height =1.1 Volume =7.99

Length =3.2 Width =2.3 Height =3.2 Volume =23.55

9)

**Max Scorer**

  Write a program to display the name of the player who has scored the maximum runs in a cricket tournament.

The player name and number of runs scored by the player are to be stored in a HashMap<String, Long>.

* Key = playerName of type String
* Value = runs of type Long

**Sample Input and Output:**

Enter the number of players

**4**

Enter the details of the player 1

**Mathew Hayden**

**572**

Enter the details of the player 2

**Adam Gilchrist**

**495**

Enter the details of the player 3

**AB de Villiers**

**465**

Enter the details of the player 4

**Suresh Raina**

**434**

Mathew Hayden

10)

**Player Details (ArrayList of objects)**

Write a program to read and display a list of player details in a specified format.   
   
Create a class named Player with the following private member variables / attributes

|  |  |
| --- | --- |
| **Data Type** | **Variable Name** |
| String | name |
| String | country |
| String | skill |

Include appropriate getters, setters and constructors.

 Include a 3-argument constructor with arguments name, country and skill.    
   
Override the toString() method to display the player details in the format specified in the output.

 String.format("%-15s %-15s %-15s", name, country, skill);    
   
Create a class named PlayerBO and include the following methods

|  |  |  |
| --- | --- | --- |
| **No** | **Method Name** | **Method Description** |
| 1 | void displayAllPlayerDetails(ArrayList playerList) | In this method, display the details of all players. |

Create a class Main

* Get inputs from the user.
* For each player, set the user inputs to the instance of Player and add the Player instance to the ArrayList.
* Invoke displayAllPlayerDetails(playerList).

**Note**: The statement "Player Details" in the output is displayed in the method inside the BO class.

**Sample Input and Output:**  
Enter the number of players    
3    
Enter the player name    
MS Dhoni    
Enter the country name    
India    
Enter the skill    
All Rounder    
Enter the player name    
Suresh Raina    
Enter the country name    
India    
Enter the skill    
All Rounder    
Enter the player name    
Michael Hussey    
Enter the country name    
Australia    
Enter the skill    
Batsman    
Player Details   
MS Dhoni                 India             All Rounder   
Suresh Raina       India             All Rounder    
Michael Hussey Australia   Batsman

11)

**Sort the Scores**

Write a program to sort the runs scored by a given player in IPL.

Store the scores scored an ArrayList.

**Input Format:**

First line of the input is an integer “n” that corresponds to the number of matches played by the player.

Next “n” lines contains an integer in each line that corresponds to the runs scored by the IPL player in each of the “n” matches.

**Output Format:**

Output should print the runs scored by the player in sorted order, in “n” lines.

**Sample Input:**

6

101

78

90

59

77

67

**Sample Output:**

59

67

77

78

90

101

12)

**Calculate Sum and Average**

Write a program to find the total points and the average points scored by a team in “n” matches.

Store the scores scored by the team in an ArrayList.

**Input Format:**

First line of the input is an integer “n” that corresponds to the number of matches played by the team.

Next “n” lines contains an integer in each line that corresponds to the points scored by the team in each of the “n” matches.

**Output Format:**

Output should print in the first line the integer that gives the total points scored by the team.

In the second line, print a float value that gives the average points.

**Note:**

* Calculate the total points scored by the team using enhanced for loop.
* Calculate the average points scored by them(total points to be calculated without using enhanced for loop).

**Sample Input :**

5

37

29

31

27

33

**Sample Output :**

157

31.40

13)

**Age Validator**

Write a program that accepts the age of the user as input. When the age of the user is less than 19, a custom exception named **InvalidAgeException**is thrown.

Use exception handling mechanisms to handle this exception.

**Sample Input and Output 1:**

Enter your age

6

Exception occured: InvalidAgeException: not valid

**Sample Input and Output 2:**

Enter your age

34

welcome to vote

14)

**Divide 2 Numbers**

Write a program that accepts 2 integers a and b as input and finds the quotient of a/b.

This program generates DivideByZeroException when the denominator is zero. Use exception handling mechanisms to handle this exception. In the catch block, print the message as shown in the sample output.

 Also illustrate the use of finally block. Print the message “Inside finally block”.

**Sample Input and Output 1:**

 Enter the 2 numbers

**5**

**2**

The quotient of 5/2 = 2

**Inside finally block**

**Sample Input and Output 1:**

  Enter the 2 numbers

**5**

**0**

**DivideByZeroException caught**

**Inside finally block**

15)

**Player Details**

* Write a program to read and display the player details to the user.
* Display the following details.
* **Cricket Player:**
* 1. Player Name
* 2. Team Name
* 3. No of matches
* 4. Total Runs Scored
* 5. No of wickets taken
* **Hockey Player:**
* 1. Player Name
* 2. Team Name
* 3. No of matches
* 4. Position
* 5. No of goals taken
* 1. Create an interface IPlayerStatistics
* - Add a method with the following prototype    
        public void displayPlayerStatistics()
* 2. Create a base abstract class Player
* - include protected data members: name, teamName, noOfMatches    
  - include 3-argument constructor with following arguments:  name, teamName, noOfMatches.
* 3. Create CricketPlayer that extends Player class and implements IPlayerStatistics
* - include private data members: totalRunsScored, noOfWicketsTaken
* - include 5-argument constructor, with the following arguments:  name, teamName, noOfMatches, totalRunsScored,noOfWicketsTaken.
* - Call the super class constructor to initialize name, teamName, noOfMatches.
* - implement the interface method public void displayPlayerStatistics() to display the player details.
* 4. Create HockeyPlayer that extends Player class and implements IPlayerStatistics
* - include private data members: position, noOfGoals
* - include 5-argument constructor, with the following arguments:  name, teamName, noOfMatches, position, noOfGoals.
* - Call the super class constructor to initialize name, teamName,noOfMatches.
* - implement the interface method public void displayPlayerStatistics() to display the player details.
* 5. Create a Main class with main method to test above classes.
* - Based on input from user either create instance of the CricketPlayer or HockeyPlayer class and assign to the reference of IPlayerStatistics.
* - Call displayPlayerStatistics() method to display the details of the player.

**Sample Input and Output 1:**  
   
Menu   
1.Cricket Player Details   
2.Hockey Player Details   
Enter choice   
**1**   
Enter player name   
**Ravichandran Ashwin**  
Enter team name   
**Chennai Super Kings**   
Enter number of matches played   
**86**  
Enter total runs scored   
**185**  
Enter total number of wickets taken   
**89**   
Player Details   
Player name : Ravichandran Ashwin   
Team name : Chennai Super Kings   
No of matches : 86   
Total runsscored : 185   
No of wickets taken : 89   
**Sample Input and Output 2:**   
1.Cricket Player Details    
2.Hockey Player Details    
Enter choice    
**2**  
Enter player name    
**Yuvraj Walmiki**  
Enter team name    
**Delhi WaveRiders**  
Enter number of matches played    
**34**  
Enter the position    
**Forward**  
Enter total number of goals taken    
**9**  
Player Details    
Player name : Yuvraj Walmiki    
Team name : Delhi WaveRiders    
No of matches : 34    
Position : Forward    
No of goals taken : 9

16)

**Compare Product Details**

Create a class named Product with the following private member variables.

* id of type Long
* productName of type String
* supplierName of type String

Include appropriate getters and setters.

Include a 3-argument constructor and a default constructor.

Include a method named equals that overrides the equals() of Object class to check the equality of Product Objects.

Two products are considered equal only when all its attribute values are same.

Create another class and write a main method to read the input and display the output as listed in the sample Output/Input.   
Create a method “void display()” in Main class to display the product details.

**Sample Input and Output 1 :**

Enter the product id

**1**

Enter the product name

**Printer**

Enter the supplier name

**HP**

Product Id is 1

Product Name is Printer

Supplier Name is HP

Enter the product id

**1**

Enter the product name

**Printer**

Enter the supplier name

**HP**

Product Id is 1

Product Name is Printer

Supplier Name is HP

The two products are the same

**Sample Input and Output 2 :**

Enter the product id

**1**

Enter the product name

**Printer**

Enter the supplier name

**HP**

Product Id is 1

Product Name is Printer

Supplier Name is HP

Enter the product id

**1**

Enter the product name

**Printer**

Enter the supplier name

**Wipro**

Product Id is 1

Product Name is Printer

Supplier Name is Wipro

The two products are different

17)

**Display Product Details**

Create a class named Product with the following private member variables.

* id of type Long
* productName of type String
* supplierName of type String

Include appropriate getters and setters.

Include a 3-argument constructor and a default constructor.

Override the toString() method defined in the Object class. Display the details of the product in this method as shown in the sample output.

Create another class and write a main method to read the input and display the output as listed in the sample Output/Input.

Invoke the getClass() method from main to retrieve the Class name.

**Sample Input and Output :**

Enter the product id

**1**

Enter the product name

**Printer**

Enter the supplier name

**HP**

1 : Printer : HP

Invoking getClass() method : class Product

18)

**Card Details**

Write a program to read and display the card details. A card can be any one of the two types, either Payback or Membership.

Based on the type of card, the kind of details to be displayed varies. Refer details below.

**Payback Card:**

1. Card Number

2. Points Earned

3. Total Amount

**Membership Card:**

1. Card Number

2. Rating

 Create an abstract class named Card with the following protected attributes / member variables.

* String holderName;
* String cardNumber;
* String expiryDate;

Include appropriate getters and setters.

Include 3-argument constructor, the order of the arguments is holderName, cardNumber, expiryDate.

  Create a class named MembershipCard derived from Card. Include the following private attributes / member variables.

* Integer rating

Include appropriate getters and setters.

Include 4-argument constructor, the order of the arguments is holderName, cardNumber, expiryDate, rating.

  Create a class named PaybackCard derived from Card. Include the following private attributes / member variables.

* Integer pointsEarned;
* Double totalAmount;

Include appropriate getters and setters.

Include a 5-argument constructor, the order of the arguments is holderName, cardNumber, expiryDate, pointsEarned, totalAmount.

  Create another class called Main. In the method, create instances of the above classes and test the above classes.

**Note**: The card details are entered separated by a ‘|’.

**Sample Input and Output 1:**

  Select the Card

1.Payback Card

2.Membership Card

**1**

Enter the Card Details:

**Anandhi|12345|14/01/2020**

Enter points in card

**1000**

Enter Amount

**50000**

Anandhi's Payback Card Details:

Card Number 12345

Points Earned 1000

Total Amount 50000.0

**Sample Input and Output 2:**

  Select the Card

1.Payback Card

2.Membership Card

**2**

Enter the Card Details:

**Collin|45678|20/11/2021**

Enter rating in card

**10**

Collin's Membership Card Details:

Card Number 45678

Rating 10

19)

**Cricket Commentary Automation**

Hope you would have used a cricket website that provides a ball by ball text commentary. Write a program to automate the text messages for each delivery.

Initially we must automate the below two display variations:

Option 1: Batsman and Bowler details of the delivery

Option 2: Number of runs scored in the delivery

Based on user’s input, either the Option 1 or Option 2 details will be displayed to the user.

**Note:**

Create a class named Delivery.

There are no attributes in this class.

 Include the following methods in the Delivery class.

* void displayDeliveryDetails(String bowler, String batsman) --- In this method, print the last names of the bowler and batsman of that particular delivery.
* void displayDeliveryDetails(Long runs) --- In this method, display the run details of that delivery. If the number of runs scored in that delivery is 6 or 4 then display “It is a Sixer.” or “It is a boundary.” along with the number of runs, else print only the number of runs.

Create a Main class to read the user inputs and invoke the displayDeliveryDetails() with appropriate parameters.

**Sample Input and Output 1:**  
Menu

1.Player details of the delivery

2.Run details of the delivery

**1**

Enter the bowler name

**Ravichandran Aswin**

Enter the batsman name

**Virat Kohli**

 Player details of the delivery:

Bowler : Ashwin

Batsman : Kohli

**Sample Input and Output 2:**

Menu

1.Player details of the delivery

2.Run details of the delivery

**2**

Enter the number of runs

**2**

Number of runs scored in the delivery : 2

**Sample Input and Output 3:**

Menu

1.Player details of the delivery

2.Run details of the delivery

**2**

Enter the number of runs

**4**

Number of runs scored in the delivery : 4

It is a boundary.

20)

**Area of a Shape (Runtime Polymorphism)**

  We are enhancing the Area of a Shape to support calculation of area for Hexagon.   
  
Area of a Hexagon needs to be calculated, but the formula to calculate area is not known at the time of implementation. Hence, we are required to create the class Hexagon and not implement any method for calculating the area.

Copy and paste all the classes available in the previous “Area of a Shape” program. Create a new class named Hexagon as per the specifications mentioned below.

**Note:**

Create a class called **Hexagon** that extends **Shape**

**Data members:**

side – of type Integer.

**Constructor:**

Create a constructor that initializes the side. (1-argument constructor).

Call the super class constructor to initialize the shapeName as "Hexagon".

Include appropriate getters and setters in all the given classes.

Create a class Main to capture the input details from the user and display the calculated area to the user. 

**Sample Input and Output 1:**

1. Rectangle

2. Square

3. Circle

4. Hexagon

Area Calculator --- Choose your shape

**4**

Enter Side:

**20**

Area of Hexagon is: 0.00

**21)**

**Area of a Shape**

Write a program to calculate the area of the given shape using a menu driven application.

Create a base class called **Shape**.

**Data members:**

protected String shapeName;

**Methods:**

calculateArea() – Return type of this method is Double. This method should return the value 0.

**Constructor:**

Create a single argument constructor that initializes the shapeName. 

Create a class called **Square** that extends **Shape**

**Data members:**

side – of type Integer.

**Methods:**

calculateArea() – calculates and returns the area of the square. The return type of this method is Double.

**Constructor:**

Create a constructor that initializes the side. (1-argument constructor).

Call the super class constructor to initialize the shapeName as "Square".

Create a class called **Rectangle** that extends **Shape**

**Data members:**

length – of type Integer.

breadth – of type Integer.

**Methods:**

calculateArea() – calculates and returns the area of the Rectangle. The return type of this method is Double.

**Constructor:**

Create a constructor that initializes the length and breadth. (2-argument constructor).

Call the super class constructor to initialize the shapeName as "Rectangle".

Create a class called **Circle** that extends **Shape**

**Data members:**

radius – of type Integer.

**Methods:**

calculateArea() – calculates and returns the area of the Circle. The return type of this method is Double.

**Constructor:**

Create a constructor that initializes the radius. (1-argument constructor).

Call the super class constructor to initialize the shapeName as "Circle".

 Include appropriate getters and setters in all the given classes.

Create a class **Main** to capture the input details from the user and display the calculated area to the user.   
**Sample Input and Output 1:**

1. Rectangle

2. Square

3. Circle

Area Calculator --- Choose your shape

**1**

Enter length and breadth:

**100**

**40**

Area of Rectangle is:4000.00

Sample Input and Output 2:

1. Rectangle

2. Square

3. Circle

Area Calculator --- Choose your shape

**2**

Enter side:

**20**

Area of Square is:400.00

**Sample Input and Output 3:**

1. Rectangle

2. Square

3. Circle

Area Calculator --- Choose your shape

**3**

Enter Radius:

**5**

Area of Circle is:78.54

**22)**

**Account Transactions**

Write a program to allow users to perform the following transactions on their bank account.

* Deposit
* Withdraw

Display the balance amount after the completion of each transaction.

**Note:**

* Create a class called Account with 2 private member variables
* accountNumber of type String
* balance of type int
* Include 2 argument constructor.
* Include getter / setter method for accountNumber.
* Include only getter for balance.
* Create a method deposit() based on below method signature. Add transactionAmount to the balance instance variable.

public void deposit(int transactionAmount)

* Create another method withdraw() which reduces the balance amount based on the transactionAmount. If the balances go below zero after withdrawal, then the transaction should not be performed, and the earlier balance should be retained. This method displays insufficient balance when the balance is about to go below zero.

public void withdraw(int transactionAmount)

**Sample Input and Output 1:**

Enter the Account Number

**1000321**

Enter the Account Balance

**5000**

Enter 1 to deposit an amount, 2 to withdraw an amount

**1**

Enter the amount to deposit

**1000**

Your balance after the transaction is: 6000

**Sample Input and Output 2:**

Enter the Account Number

**1000321**

Enter the Account Balance

**5000**

Enter 1 to deposit an amount, 2 to withdraw an amount

**2**

Enter the amount to withdraw

**1000**

Your balance after the transaction is: 4000

**Sample Input and Output 3:**

Enter the Account Number

**1000321**

Enter the Account Balance

**5000**

Enter 1 to deposit an amount, 2 to withdraw an amount

**2**

Enter the amount to withdraw

**10000**

Insufficient Balance

Your balance after the transaction is: 5000

**23)**

**Display Trainee Details**

Write a program to read and display the Trainee details for the batch JAVA.

Read the following Trainee details from the user:

* Employee Id
* Name

Display the following details to the user:

* Employee Id
* Name
* BATCH Code

**Note:**

* Create a Class Trainee with instance variables employeeId and name with appropriate constructor and getter / setters.
* Declare and initialize static variable BATCH\_CODE in Trainee class as mentioned below.

                  private static final String BATCH\_CODE = “JAVA”;

* Create a method display() in Trainee class to display the details to the user.
* In each Java class the code should be organized in such a way the declarations are done in the order specified below. Please ensure that this order is organized in this class. static variables
  + instance variables
  + constructors
  + getters and setters
  + other methods
* Create a class Main which does the following:
  + Read the number of Trainee's details to be read
  + Read each Trainee’s employeeId and name.
  + Create an instance of Trainee class and invoke the display() method.

**Sample Input and Output:**

Enter the number of Trainees

**2**

Enter Employee Id

**969143**

Enter Name

**John**

Enter Employee Id

**969144**

Enter Name

**Priya**

969143 John JAVA

969144 Priya JAVA

**24)**

**Product Details (Constructors)**

Create a class named Product with the following private member variables.

* id of type Long
* productName of type String
* supplierName of type String

Include appropriate getters and setters.

Include a 3-argument constructor and a default constructor.

Create a method “void display()” to display the product details.

Create another class and write a main method to perform the following steps:

1. Read the input
2. Create instance of Product using 3 argument constructor.
3. Create a method “void display()” to display the product details using getter method.

**Sample Input and Output :**

Enter the product id

**1**

Enter the product name

**Printer**

Enter the supplier name

**HP**

Product Id is 1

Product Name is Printer

Supplier Name is HP

**25)**

**Product Details (Class and Instance)**

Create a class named Product with the following private member variables.

* id of type Long
* productName of type String
* supplierName of type String

Include appropriate getters and setters.

Create another class and write a main method to perform the following steps:

1. Read the input
2. Create instance of Product and set the values into Product instance using setter methods.
3. Display the output as listed in the sample Output/Input using getter methods.

**Sample Input and Output:**

Enter the product id

**1**

Enter the product name

**Printer**

Enter the supplier name

**HP**

Product Id is 1

Product Name is Printer

Supplier Name is HP