Name: Shweta Rawat

Employee ID: **902483**

**Batch: INTCDB21DW002**

**Exercise1: Profit Calculation**   
  
In order to calculate profit percentage of the Electronic gadgets sold in a store, you need to store the buying price and selling price of the gadgets.  
  
Write a program to store the buying price and selling price of the products and print the values to the console.  
  
Note:

* Create the variables buyingPrice,sellingPrice
* Assign the variables with values for buying price as 20.54 and selling price as 30.50

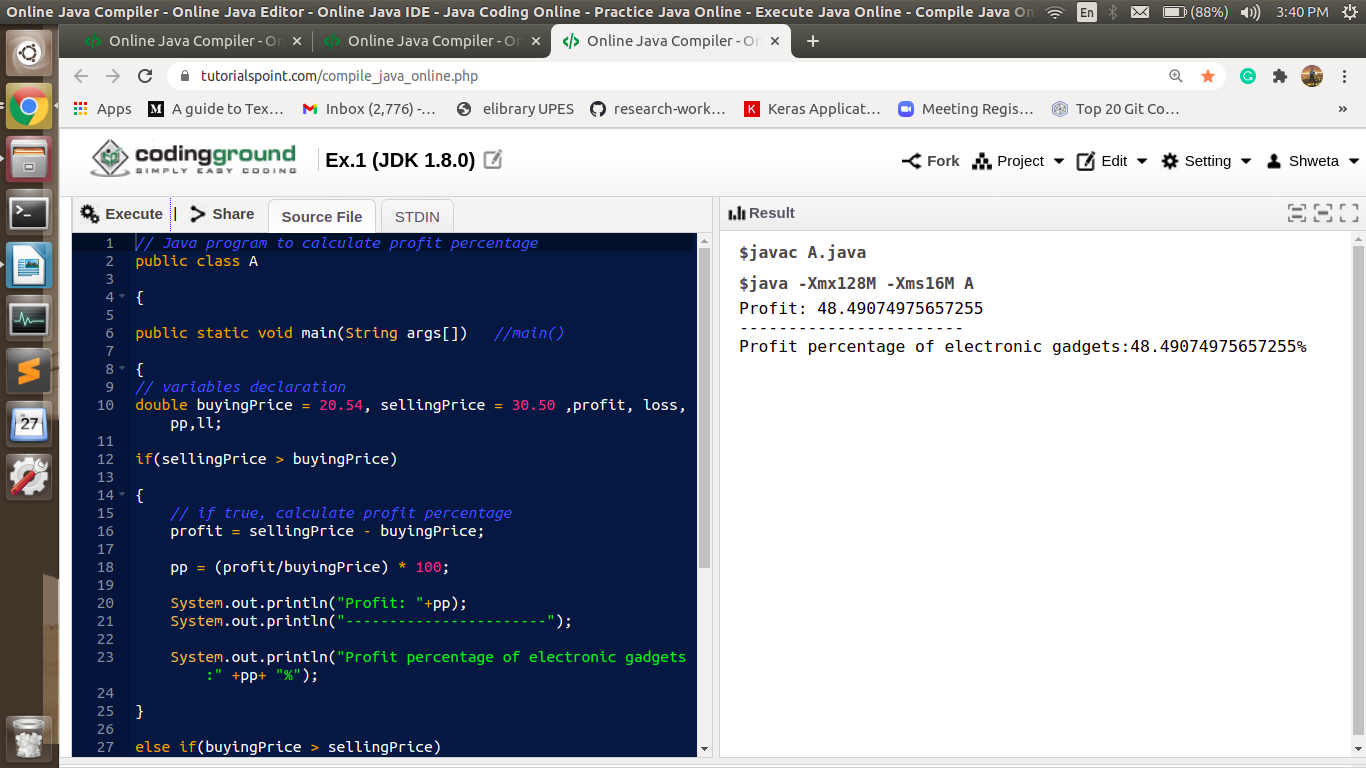
**NOTE:**Code submission not required. To be implemented using eclipse IDE and manually verified by the SME. 

**Sample Output:**

Buying price is 20.54

Selling price is 30.50

**Output:**



**Exercise2: 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 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.

 NOTE: Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

Sample Input and Output:

[All text in bold corresponds to input and the rest corresponds to 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

**Output:**



**Exercise3: 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)

NOTE: Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

Sample Input and Output Format:

All text in bold corresponds to input and the rest corresponds to output.

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

**Output:**

****

**Exercise4: Lab Allocation**

There are 3 labs in the CSE department (L1, L2 and L3) with a seating capacity of x, y and z. Find the lab which has the minimal seating capacity.

Note: Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

Input and Output Format:

Assume that x, y and z are always distinct.

Refer sample input and output.

  All text in bold corresponds to input and the rest corresponds to output.

Sample Input and Output 1:

Enter x

30

Enter y

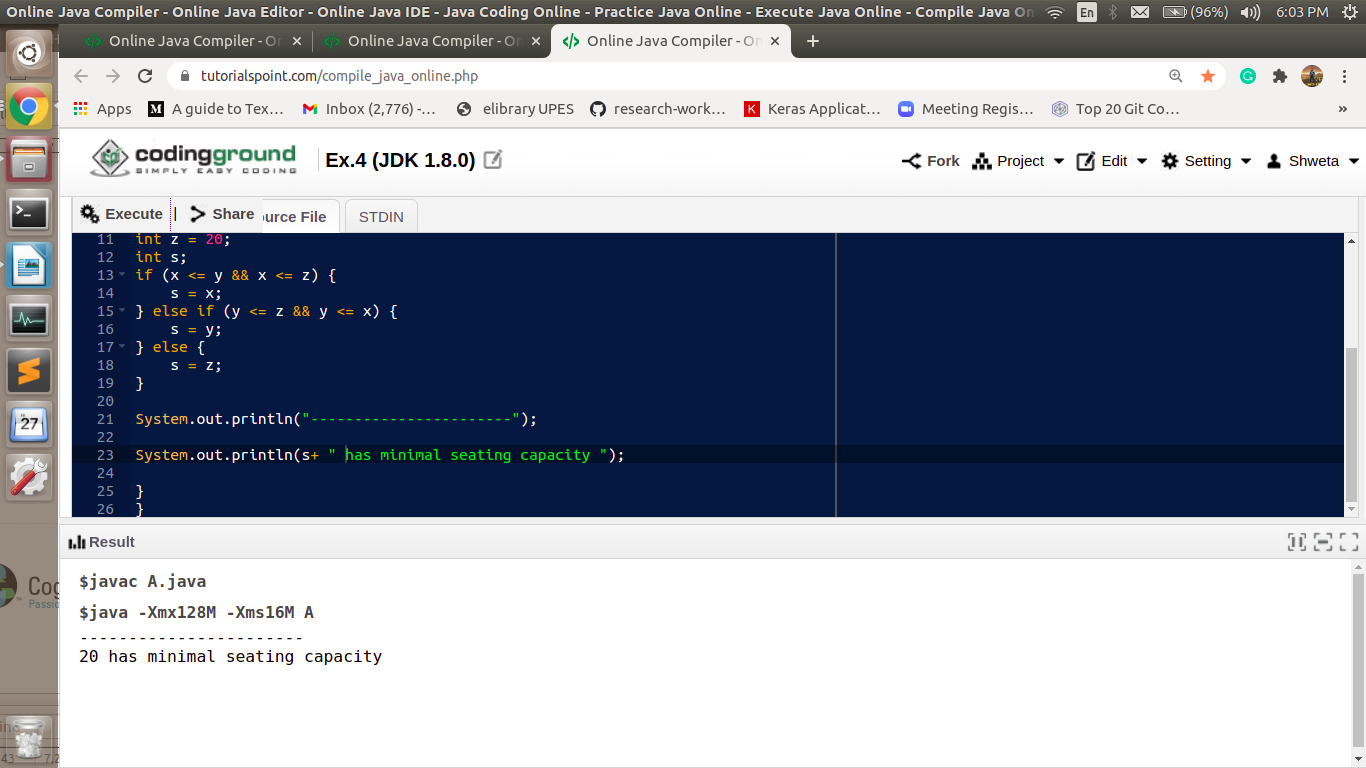
40

Enter z

20

L3 has the minimal seating capacity

**Output:**



**Exercise5: Compare 2 Integers**

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

Note: Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

Input and Output Format:

Input consists of 2 integers.

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

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

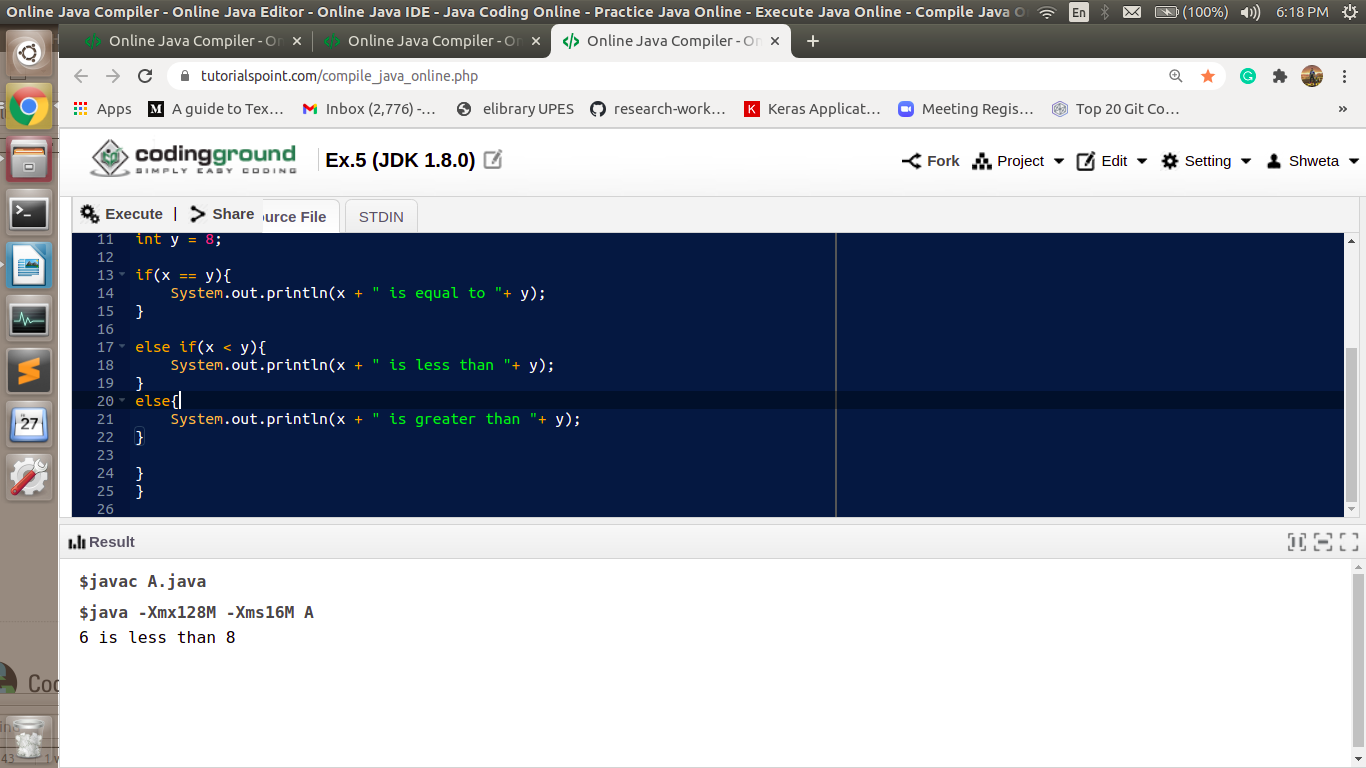
8

Enter the second number

8

8 is equal to 8

**Output:**



**Exercise6: Secure URL**

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

**Example:**

Secure URL: https://www.amazon.com/

**Input and Output Format:**

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

**Note:**Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

**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

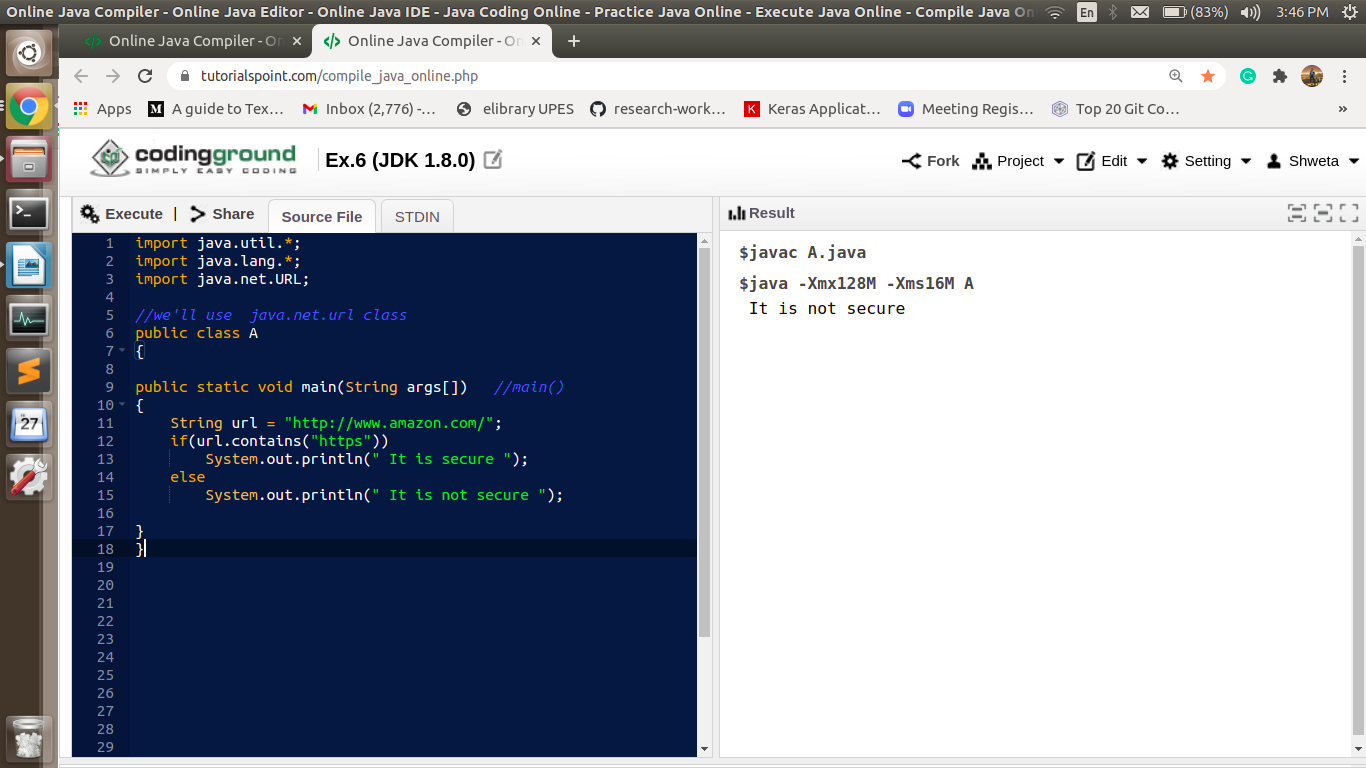
[**https://www.amazon.com/**](https://www.amazon.com/)

Enter the start string

**https**

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

**Output:**



**Exercise7: 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.

**Input and Output Format:**

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

**NOTE:** Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

**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**

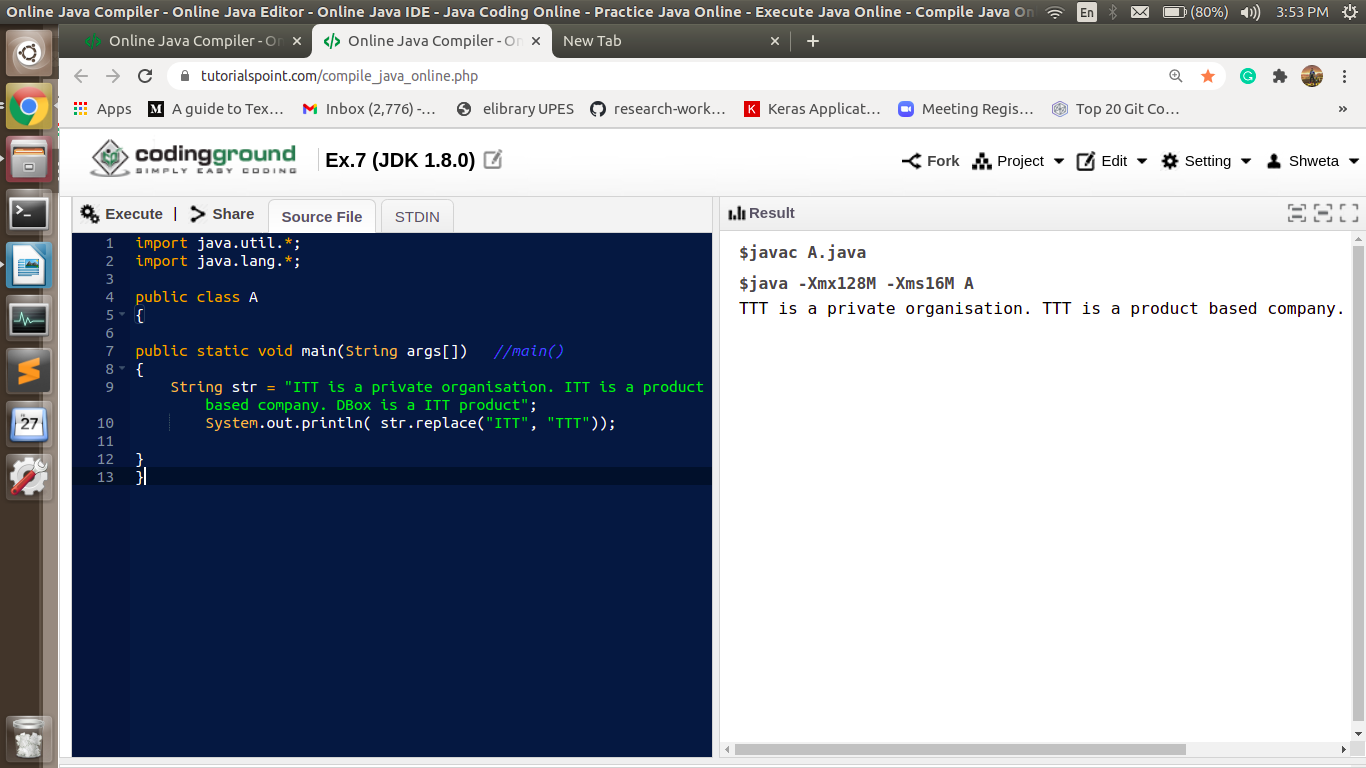
Enter the new name of the company

**TTT**

The content of the modified document is

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

**Output:**



**Exercise8: Day of the Week (Using direct Array initialization)**

Write a program to find the day of the week for the day number given. 1 should return Sun, 2 should return Mon, etc.

**Note:**  
Declare and Initialize array in a single line using curly braces.

**Input and Output Format:**

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

**Note:** Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

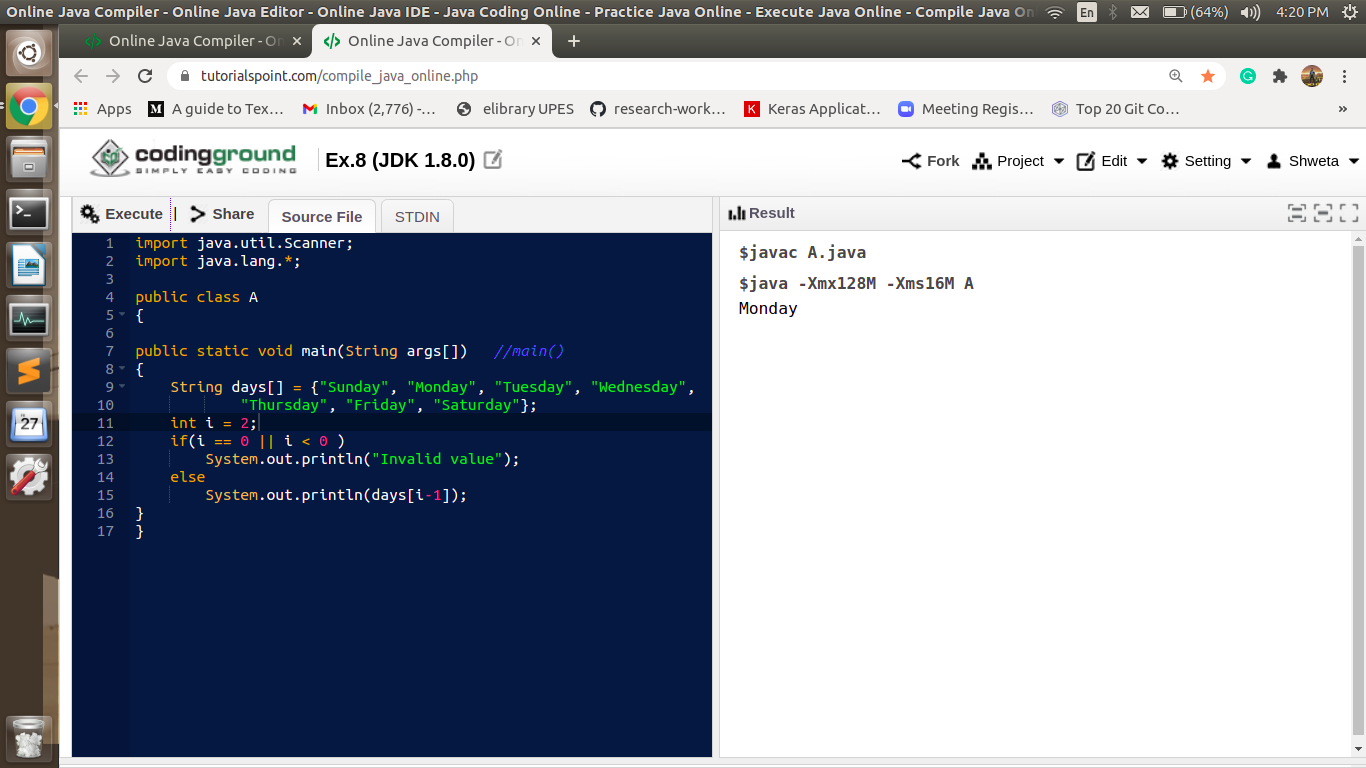
**Sample Input and Output:**

Enter the day number

3

Day of the week is Tue

**Output:**



**Exercise9:** **Day of the Week (Using new keyword)**

Write a program to find the day of the week for the day number given. 1 should return Sun, 2 should return Mon, etc.

**Note:**  
Declare and Initialize array elements using new keyword.

**Input and Output Format:**

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

**Note:**Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

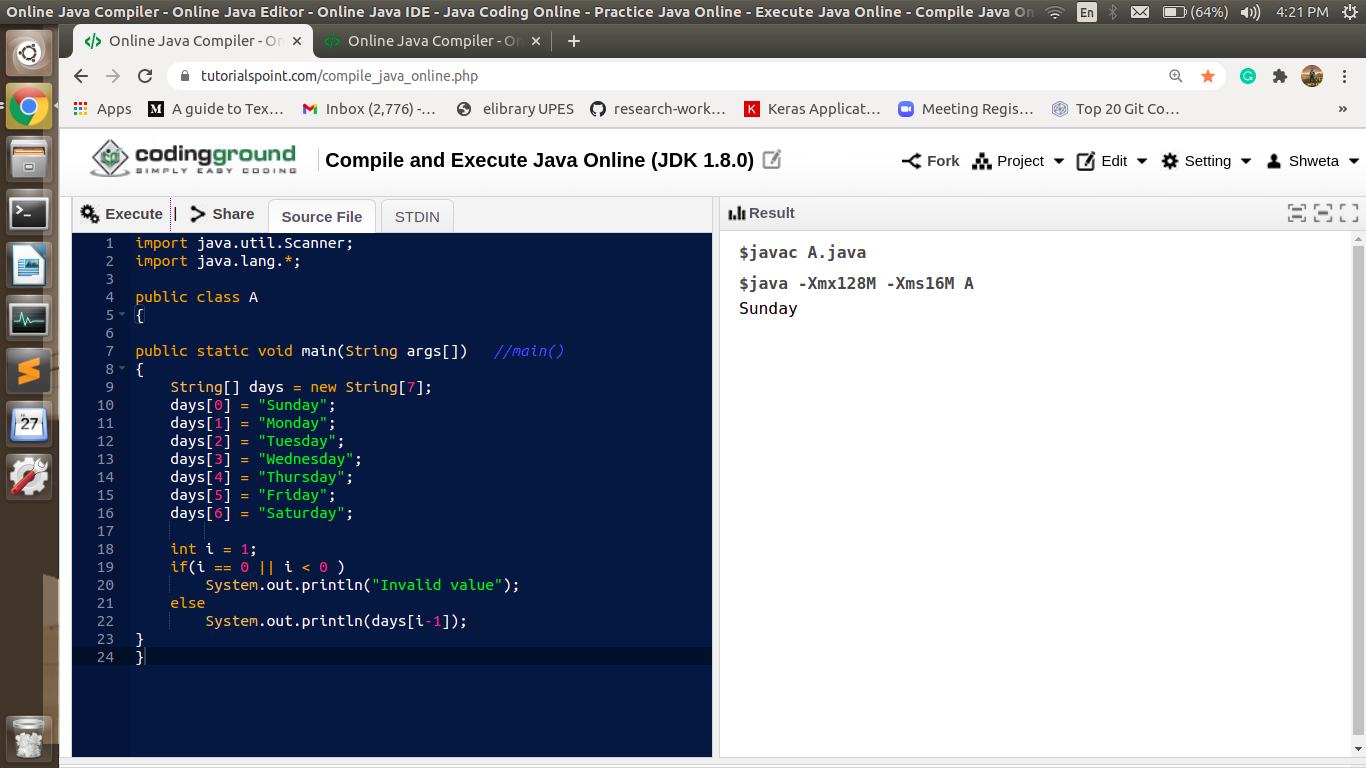
**Sample Input and Output:**

Enter the day number

3

Day of the week is Tue

**Output:**



**Exercise10: The Red Cross (for loop)**

On an event of a natural calamity, Red Cross is responsible to present the accurate details of the number of people who were rescued and sought refuge in the camps.

Write a program to facilitate Red Cross authorities record the total number of refugees from each tent of the camps given the number of tents.

**Note:**

* Use integer array to store the number of refugees present in each tent.
* Use for loop to store people count in tent and calculate the total number of people.

**Input Format:**

First line of the input is an integer value that corresponds to the number of tents in the camp.   
The next n lines of input contains the number of refugees staying in each of the tents.

**Output Format:**

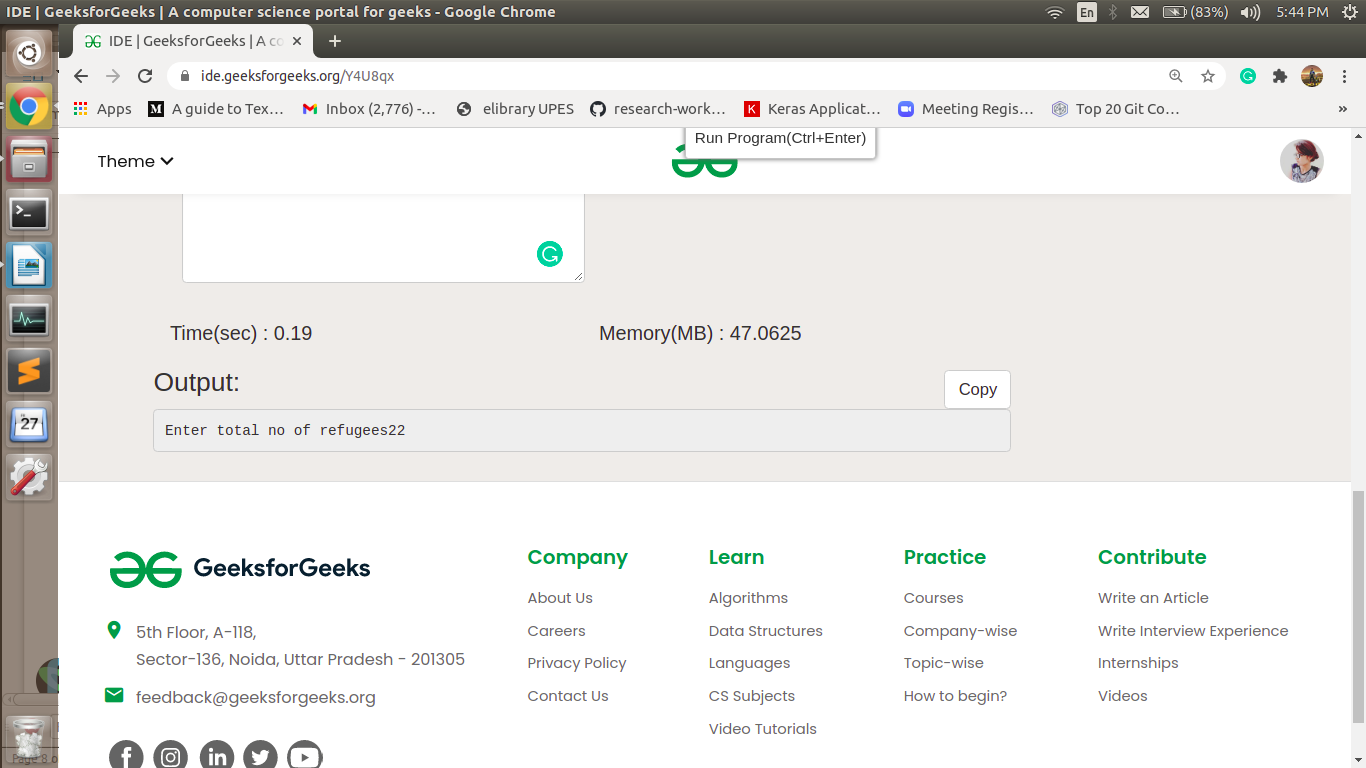
Output should print the total number of citizens present in the camp.   
Refer sample input and output for formatting specifications.   
All text in bold corresponds to input and the rest corresponds to output.

**Note:**Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

**Sample Input and Output:**

**5   
10   
12   
30   
12   
12**  
76

**Output:**



**Exercise11: The Red Cross (while loop)**

On an event of a natural calamity, Red Cross is responsible to present the accurate details of the number of people who were rescued and sought refuge in the camps.

Write a program to facilitate Red Cross authorities record the total number of refugees from each tent of the camps given the number of tents.

**Note:**

* Use integer array to store the number of refugees present in each tent.
* Use while loop to store people count in tent and calculate the total number of people.

**Input Format:**

First line of the input is an integer value that corresponds to the number of tents in the camp.

The next n lines of input contains the number of refugees staying in each of the tents.

**Output Format:**

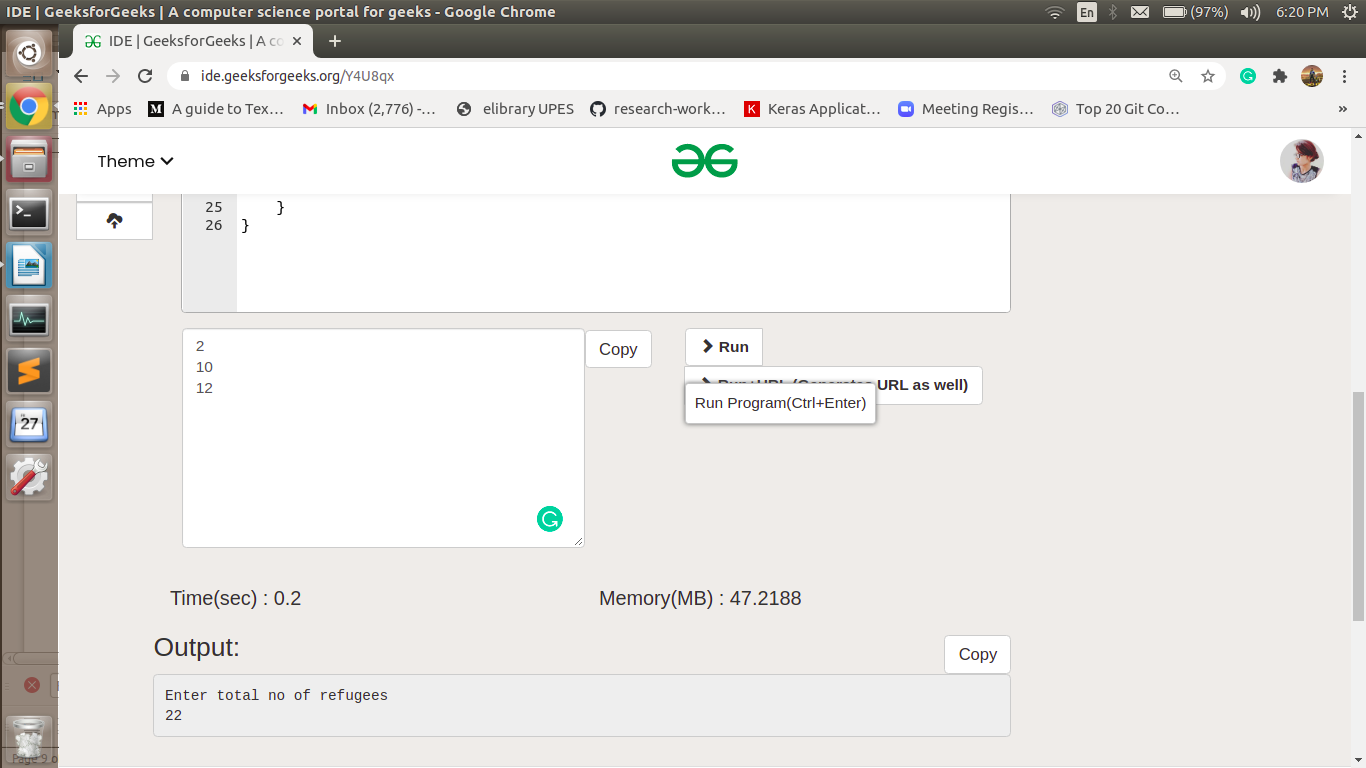
Output should print the total number of citizens present in the camp.

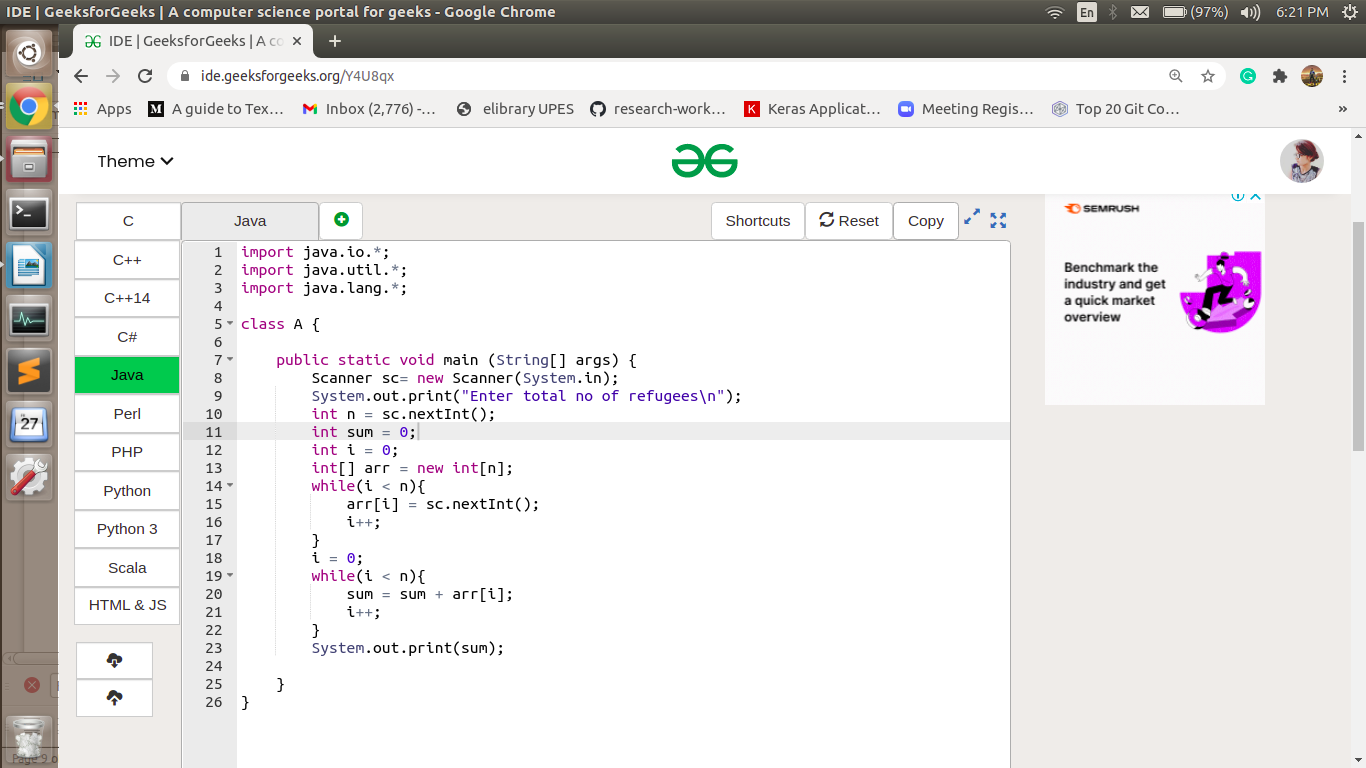
Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.   
  
**Note: C**ode submission not required. To be implemented using eclipse IDE and manually verified by the SME.

**Sample Input and Output:**  
**5   
10   
12   
30   
12   
12**  
76

**Output:**





**Exercise12: The Red Cross (for-each loop)**

On an event of a natural calamity, Red Cross is responsible to present the accurate details of the number of people who were rescued and sought refuge in the camps.

Write a program to facilitate Red Cross authorities record the total number of refugees from each tent of the camps given the number of tents.

**Note:**

* Use integer array to store the number of refugees present in each tent.
* Use for loop to store people count in tent.
* Use separate enhanced for loop to calculate the total number of people.

**Input Format:**

First line of the input is an integer value that corresponds to the number of tents in the camp.

The next n lines of input contains the number of refugees staying in each of the tents.

**Output Format:**

Output should print the total number of citizens present in the camp.

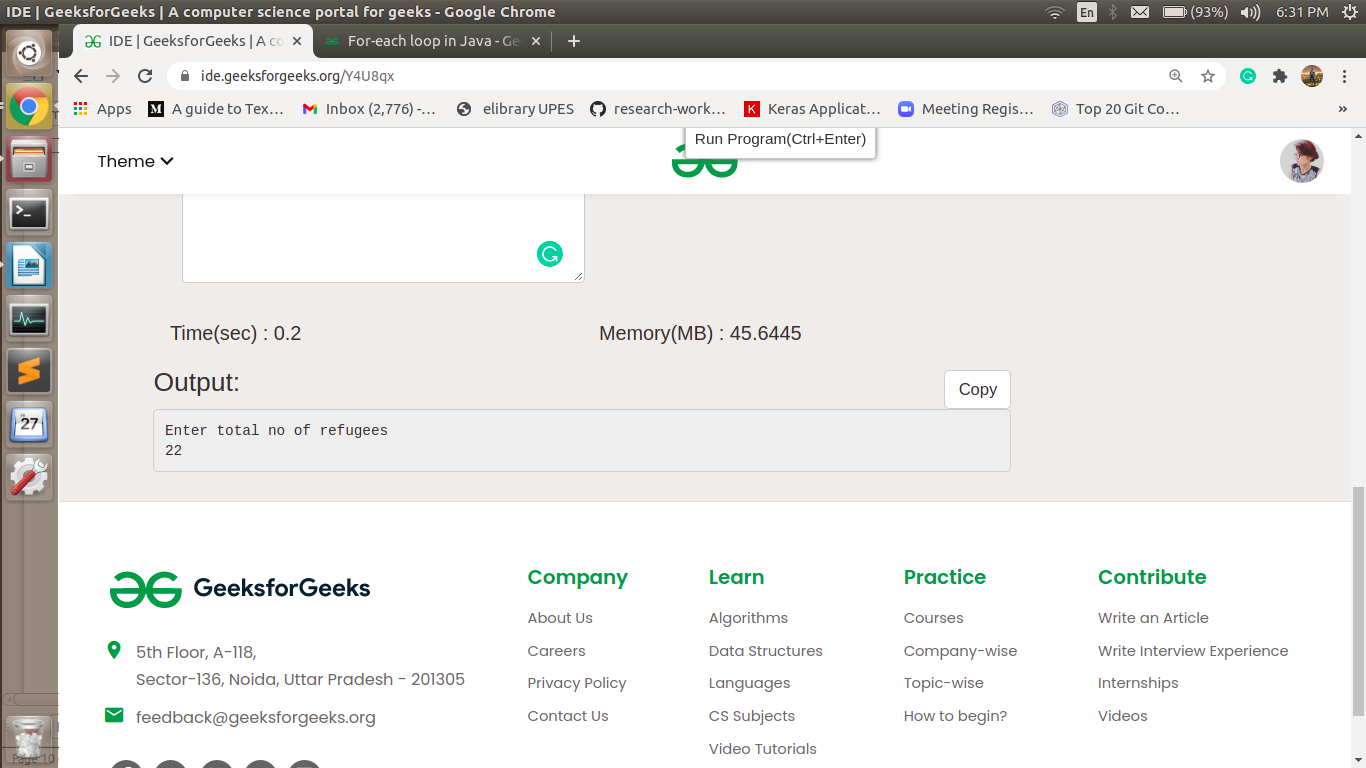
Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

**Note: C**ode submission not required. To be implemented using eclipse IDE and manually verified by the SME.

**Sample Input and Output:**  
**5   
10   
12   
30   
12   
12**  
**76**

**Output:**

****

**Exercise13: 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.

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.

NOTE: Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

Sample Input and Output:

[All text in bold corresponds to input and the rest corresponds to output]

Enter the number of dozens of toys purchased

20

Enter the price per dozen

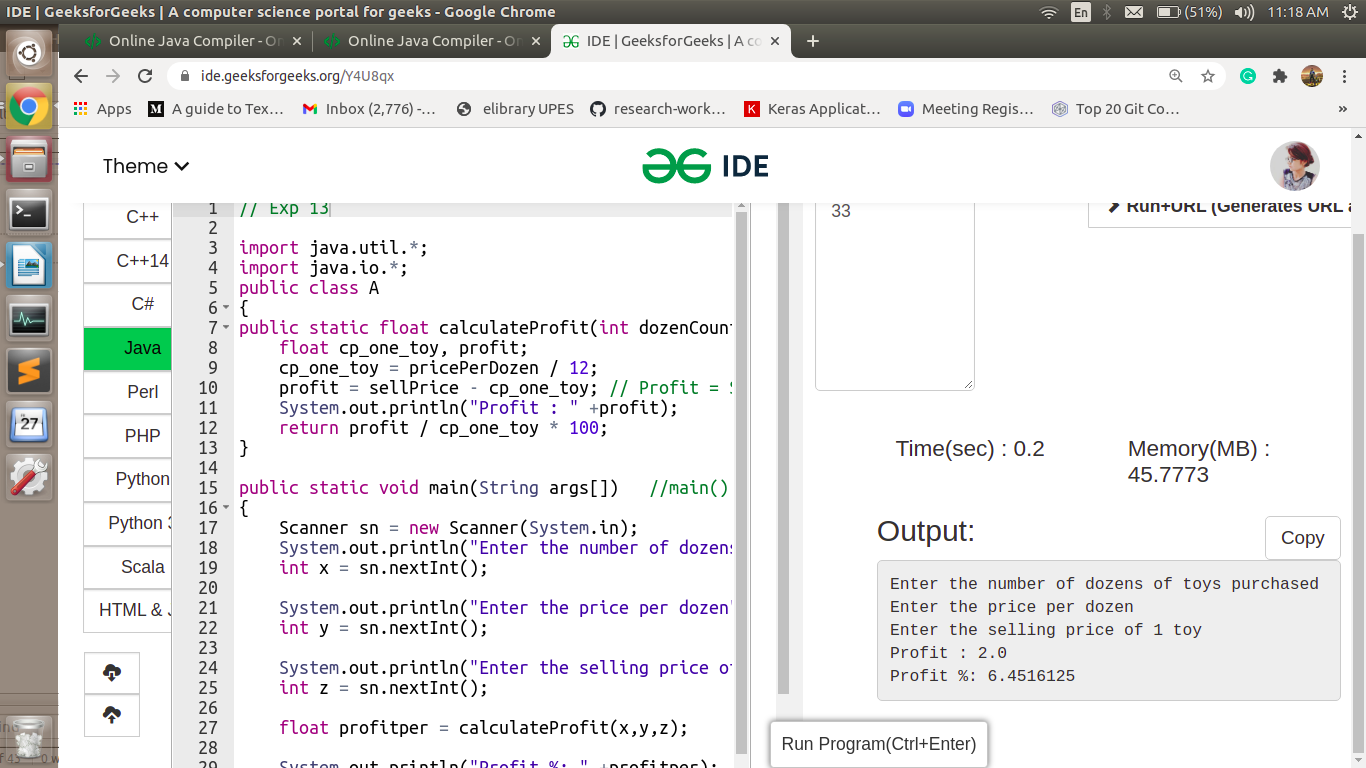
375

Enter the selling price of 1 toy

33

Sam's profit percentage is 5.60 percent

**Output:**



**Exercise14: 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.

Input and Output Format:

 Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

Note: Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

Sample Input and Output:

Enter the product id

1

Enter the product name

Printer

Enter the supplier name

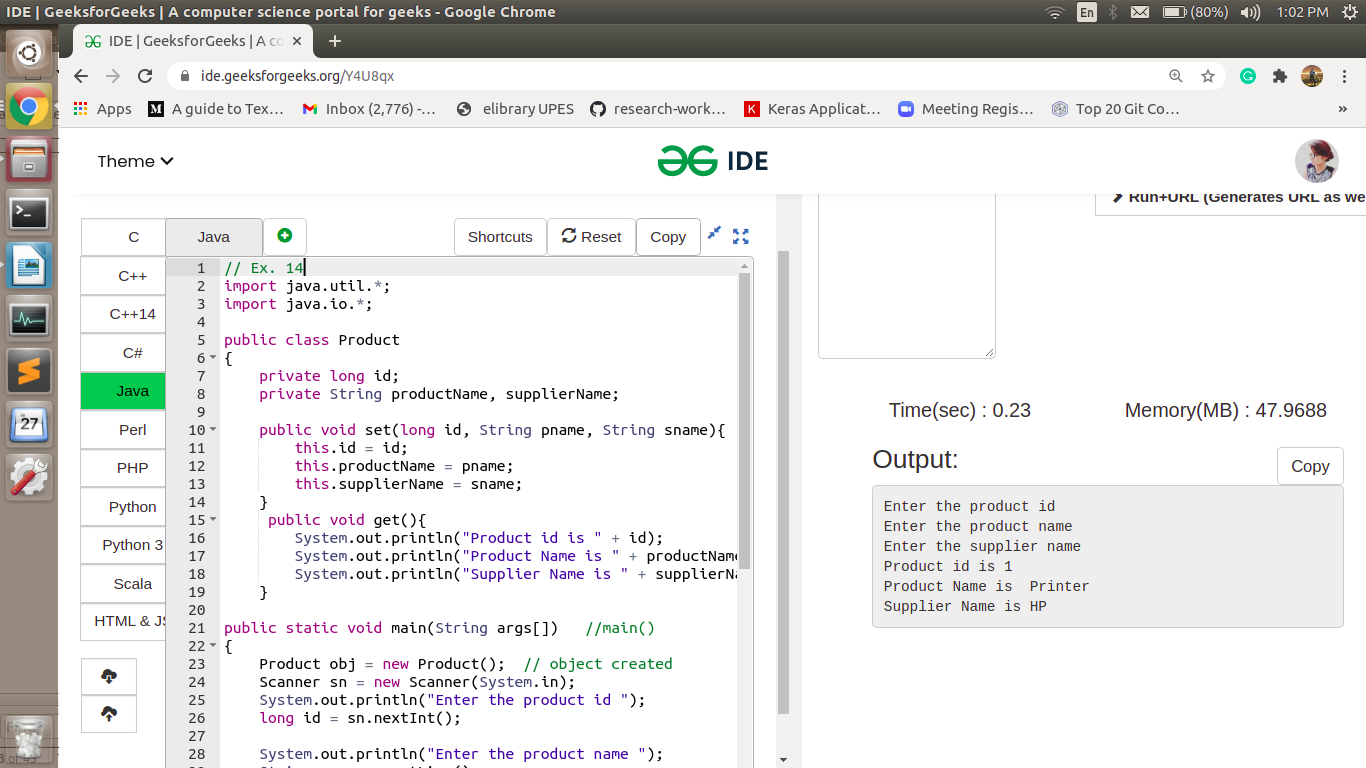
HP

Product Id is 1

Product Name is Printer

Supplier Name is HP

**Output:**

****

**Exercise15:** 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.

Input and Output Format:

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

Note: Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

Sample Input and Output:

Enter the product id

1

Enter the product name

Printer

Enter the supplier name

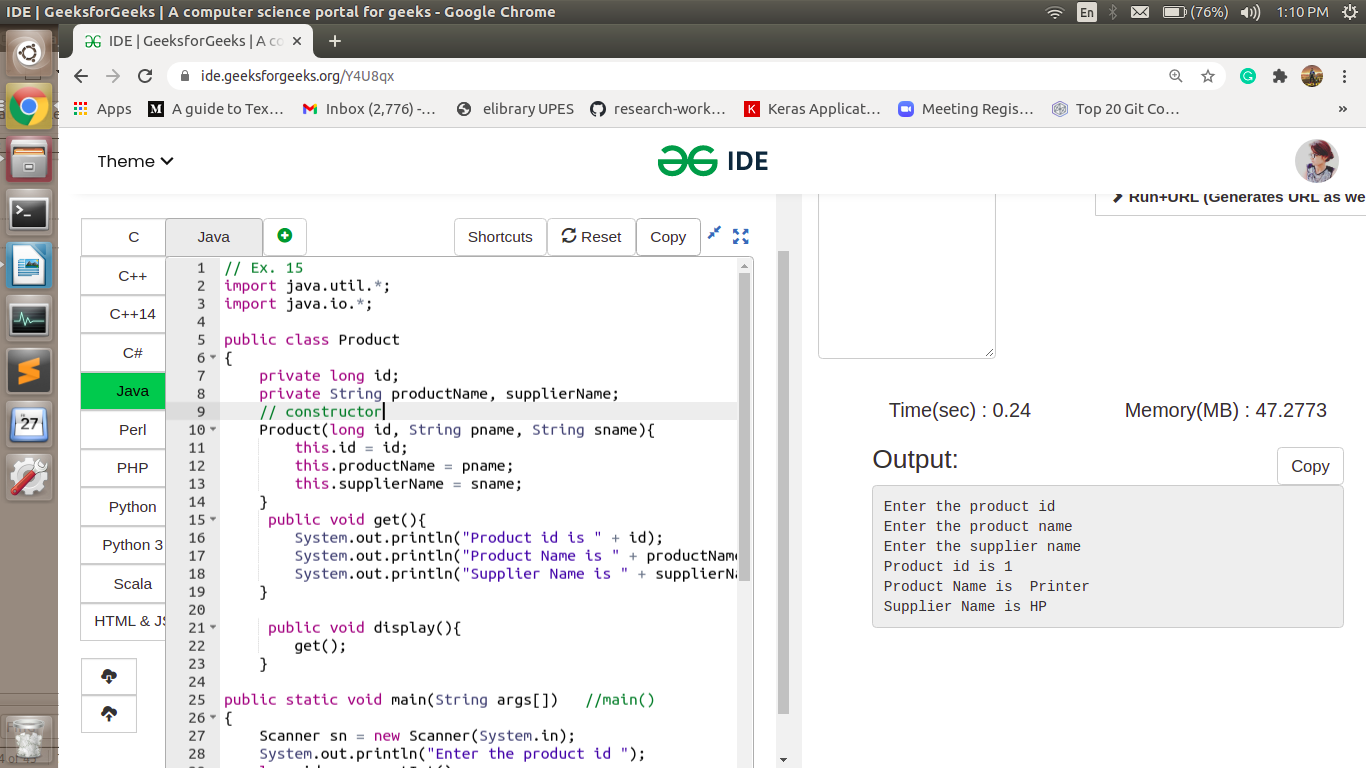
HP

Product Id is 1

Product Name is Printer

Supplier Name is HP

**Output:**



**Exercise16: Display GenC Details**

Write a program to read and display the GenC details for the batch CHNAJ19004.

Read the following GenC details from the user:

• Employee Id

• Name

Display the following details to the user:

• Employee Id

• Name

• Cohort Code

Note:

• Create a Class Trainee with instance variables employeeId and name with appropriate constructor and getter / setters.

• Declare and initialize static variable COHORT\_CODE in Trainee class as mentioned below.

private static final String COHORT\_CODE = “CHNAJ19004”;

• 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

o instance variables

o constructors

o getters and setters

o other methods

• Create a class Main which does the following:

o Read the number of GenC's details to be read

o Read each GenC’s employeeId and name.

o Create an instance of Trainee class and invoke the display() method.

Input and Output Format:

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

 NOTE: Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

Sample Input and Output:

Enter the number of GenCs

2

Enter Employee Id

969143

Enter Name

John

Enter Employee Id

969144

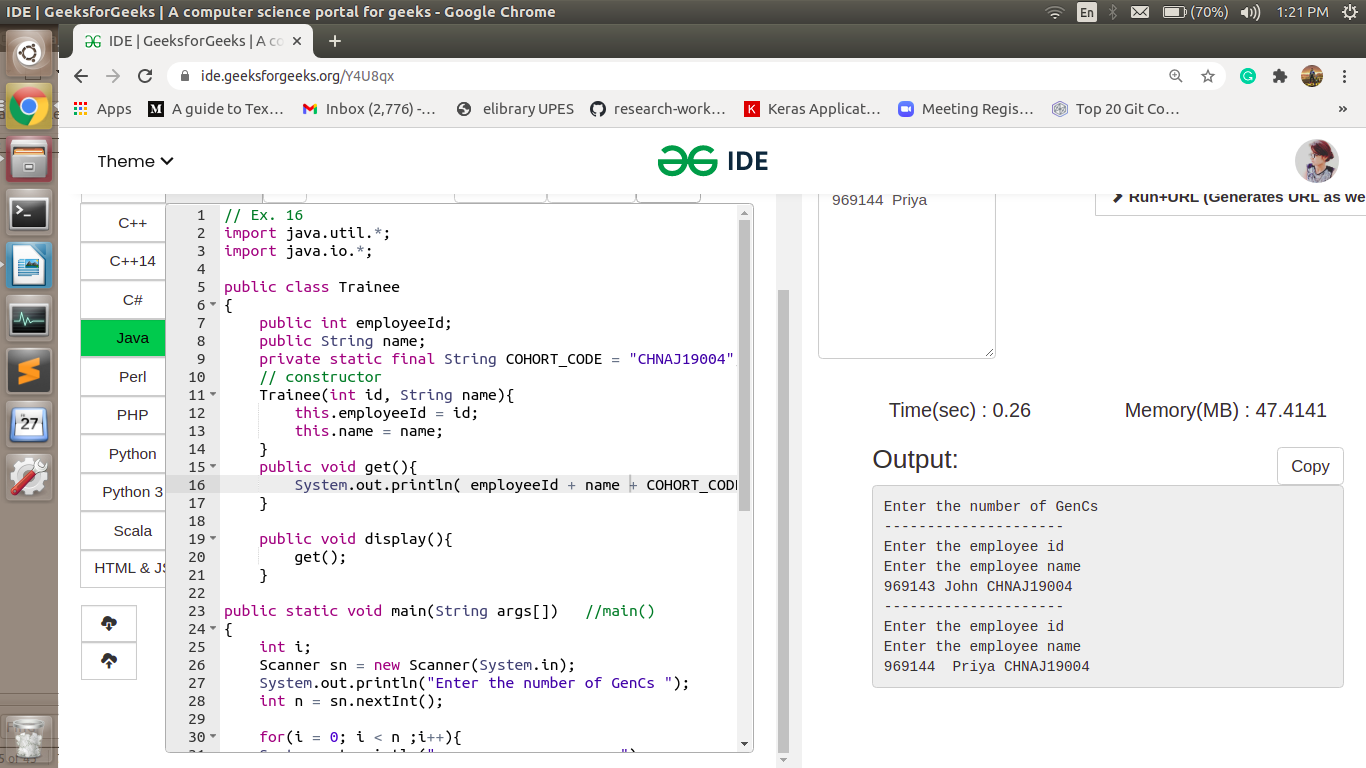
Enter Name

Priya

969143 John CHNAJ19004

969144 Priya CHNAJ19004

**Output:**



**Exercise17: 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)

**Input and Output Format:**

Refer sample input and output for formatting specifications.

**Note: C**ode submission not required. To be implemented using eclipse IDE and manually verified by the SME.

**Sample Input and Output:**

All text in bold corresponds to input and the rest corresponds to output.

**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**

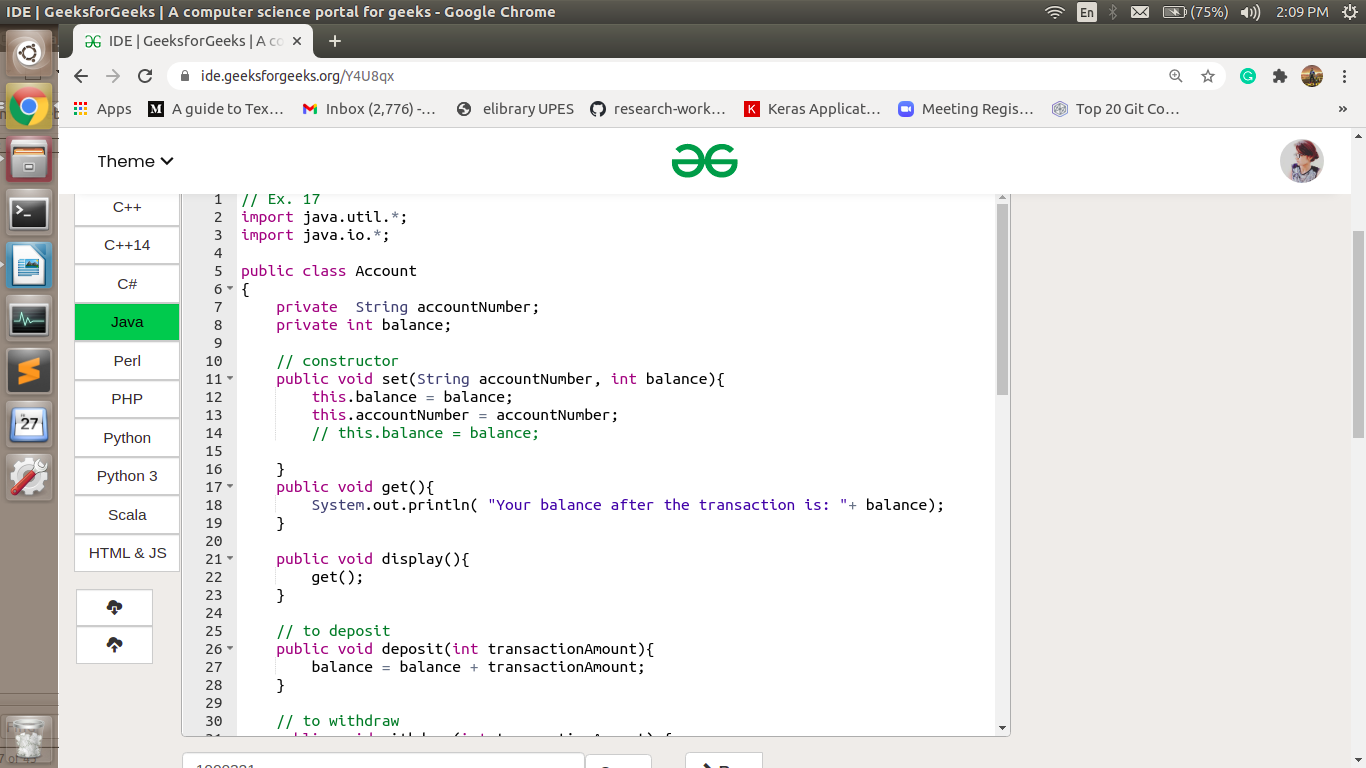
Enter the amount to withdraw

**10000**

Insufficient Balance

Your balance after the transaction is: 5000

**Output:**



**Exercise18: 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. 

**Input and Output Format:**

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.   
 Format the output with two decimal points

**Note:** Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.  
**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

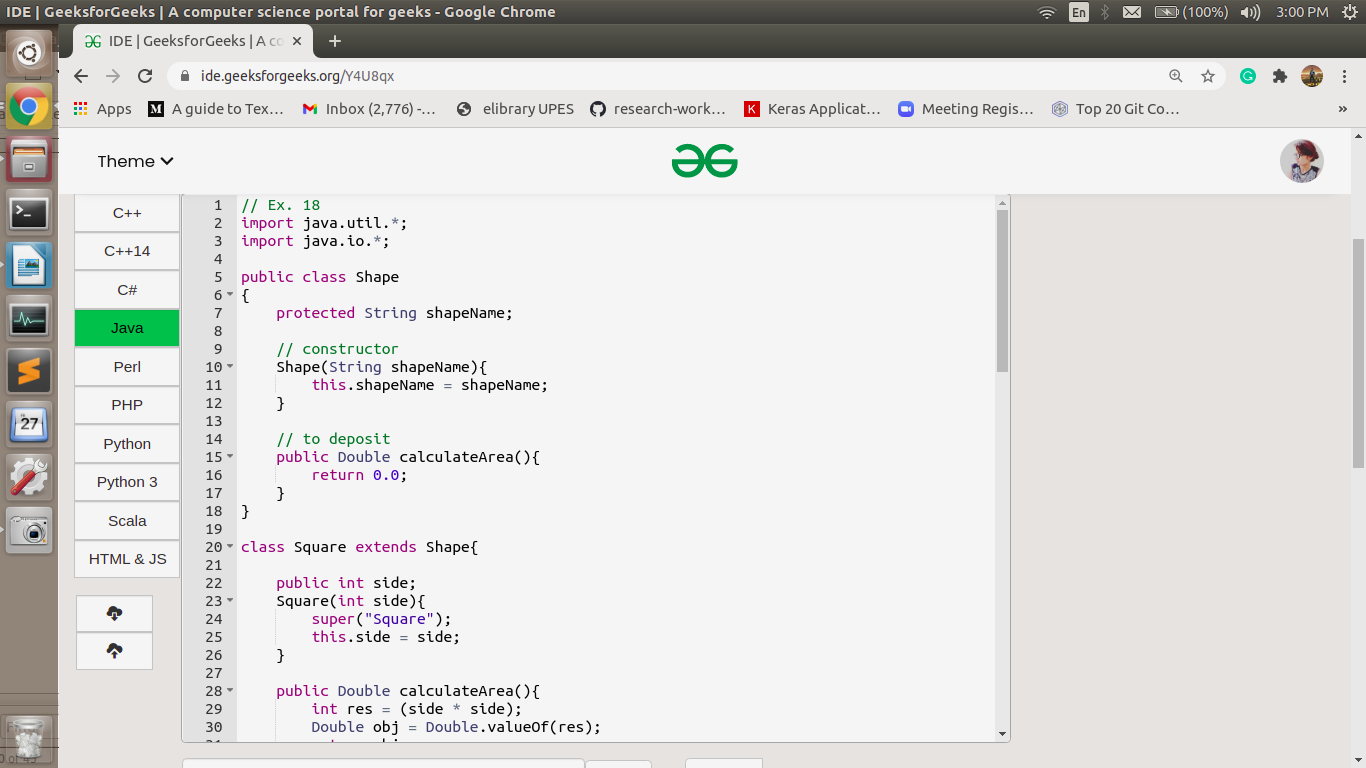
**3**

Enter Radius:

**5**

Area of Circle is:78.54

**Output:**



**Exercise19: 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. 

**Input and Output Format:**

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

Format the output with two decimal points.  
  
**Note:** Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

**Sample Input and Output 1:**

1. Rectangle

2. Square

3. Circle

4. Hexagon

Area Calculator --- Choose your shape

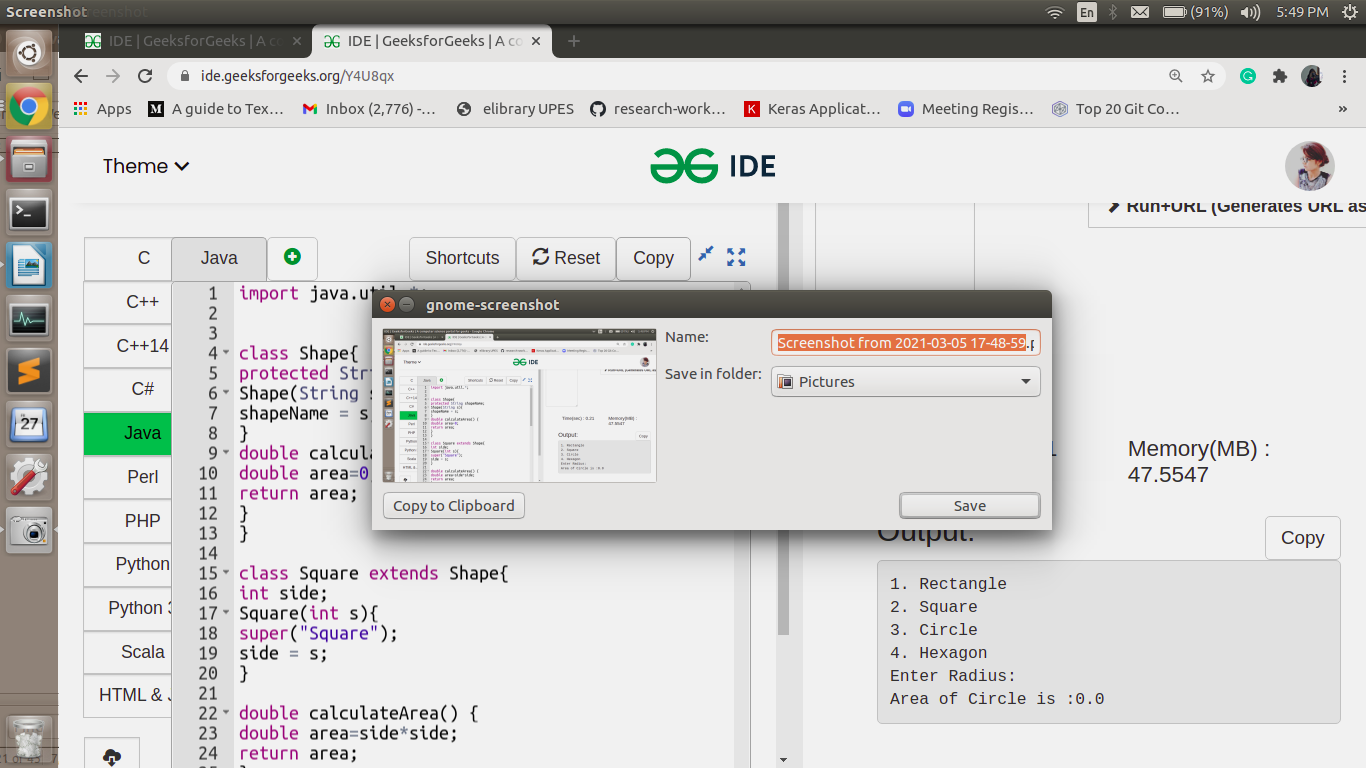
**4**

Enter Side:

**20**

Area of Hexagon is: 0.00

**Output:**



**Exercise20: 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.

**Input and Output Format:**

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

**Note:**Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.  
  
**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.

**Output:**



**Exercise21: 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 ‘|’.

**Input and Output Format:**

  Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

**Note:** Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

**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

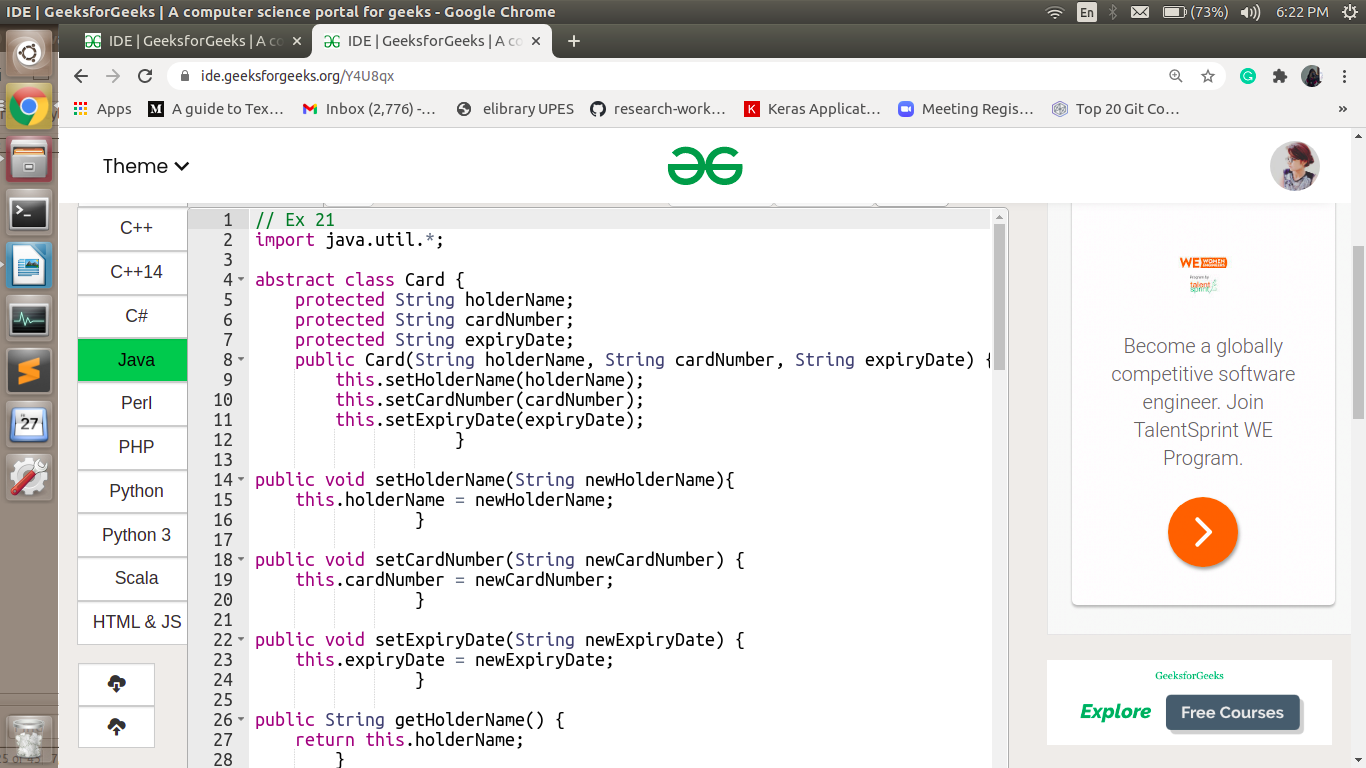
**10**

Collin's Membership Card Details:

Card Number 45678

Rating 10

**Output:**



**Exercise22**: **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. 

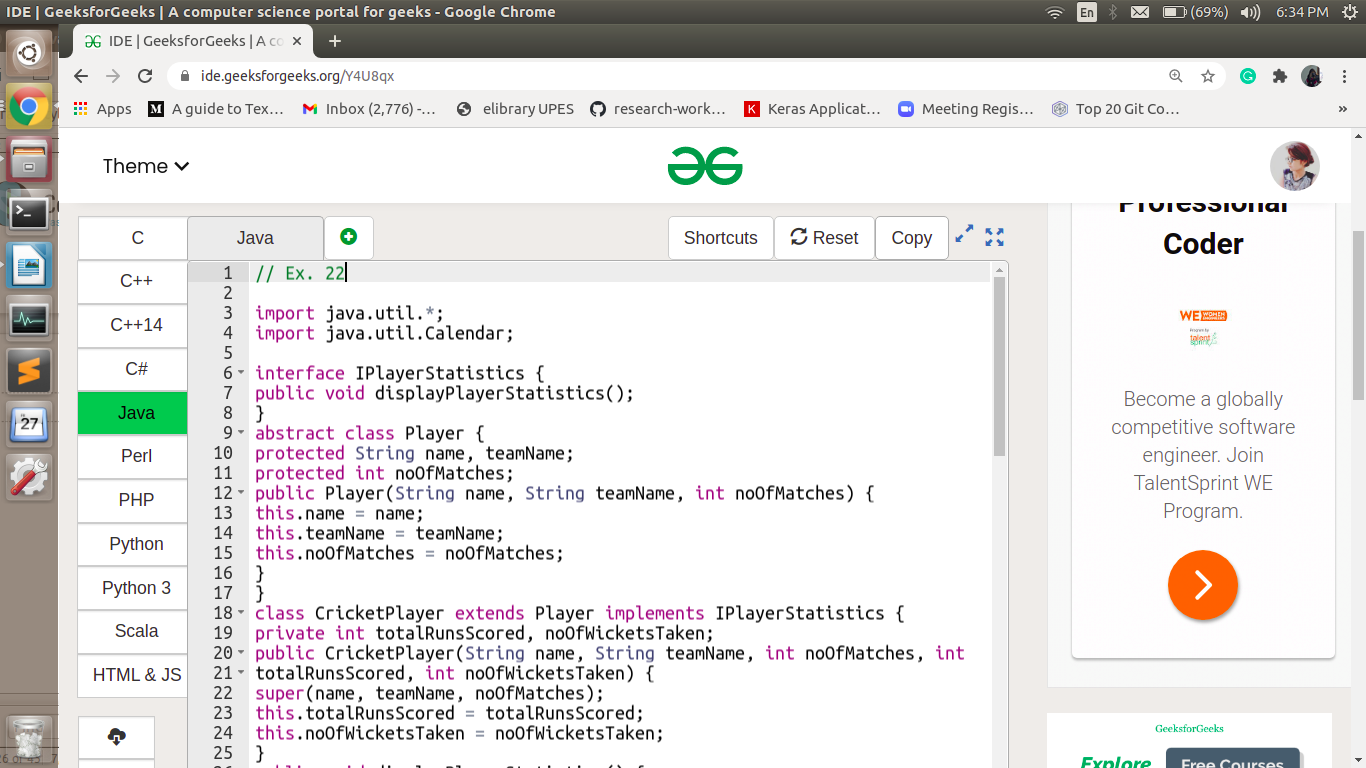
5. Create a Main class with main method to test above classes.

- 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.

**Note:**Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.  
 [All text in bold corresponds to input and the rest corresponds to output.]    
**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   
   
   
[All text in bold corresponds to input and the rest corresponds to output.]    
**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

**Output:**



**Exercise23: 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.

**Input and Output Format:**

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

**NOTE: C**ode submission not required. To be implemented using eclipse IDE and manually verified by the SME.

**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

**Exercise24: 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.

**Input and Output Format:**

 Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

**Note:** Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

**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

**Collection Framework**

**Exercise25: 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).

**Note: C**ode submission not required. To be implemented using eclipse IDE and manually verified by the SME.  
  
**Sample Input:**

5

37

29

31

27

33

**Sample Output:**

157

31.40

**Exercise26:** **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.

**Note:**Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

**Sample Input:**

6

101

78

90

59

77

67

**Sample Output:**

59

67

77

78

90

101

**Exercise27: 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).

**Input and Output Format:**

 Refer sample input and output for formatting specifications.    
All text in bold corresponds to input and the rest corresponds to output.

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

**Note:**Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.  
**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

**Exercise28: 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

**Input and Output Format:**

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

**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

**Exercise29: 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

**Advanced Java**

**Exercise30: IO - Simple File Write**

Write a java program to record the player details into the file.

Get the player details name, teamName and number of matches played from the user and write that information to a comma separated format (CSV) file.

**Output File Format:**

<name>,<teamName>,<numberOfMatches>

**Example:**

Virat Kohli,Royal Challengers Bangalore,16

**Note:**

Create a main class "Main.java"

* Read Input retails from the user
* Create a new file – player.csv
* Write the data to the file using FileWriter

**Input and Output Format:**

Get the player details name, teamName and noOfMatches from the user

[All text in bold corresponds to input and the rest corresponds to output]  

**Sample Input/Output :**

Enter the name of the player

Virat Kohli

Enter the team name

Royal Challengers Banglore

Enter the number of matches played

16

**Note:** Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

**Display Date**

Given a date in the form of string, write a program to convert the given string to date.  
Include a class **UserMainCode**with a static method **displayDate**which accepts a string. In this method display the given string in date format yyyy-MM-dd. The return type is void.  
  
 Create a Class **Main**which would be used to accepts a string and call the static method present in UserMainCode.  
  
**Input and Output Format:**  
Input consists of a string.  
Output consists of Date.  
  
Refer sample output for formatting specifications.  
  
**Sample Input 1:**  
May 1, 2016  
**Sample Output 1:**  
2016-05-01  
  
**Sample Input 2:**  
March 21, 2016  
**Sample Output 2:**  
2016-03-21  
  
**NOTE:**Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

**Exercise31: EMI Month Calculator**

Banking firm needs to calculate the loan EMI start date for its customers who have defaulted EMI payment.   
Write a program to calculate the loan EMI start date, which is 20 months before the given date.    
   
Create a class UserMainCode with a static method displayDate(), which accepts the given date as String in format yyyy-MM-dd and displays the calculated date in the format yyyy-MM-dd as shown in the sample output.

Create a Class Main, which accepts a String and call the static method present in UserMainCode.

**Note:**

Use Calendar class to calculate the required date.   
**Note:**Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.  
**Input and Output Format:**  
Input consists of a string.    
Refer sample output for formatting specifications.    
   
**Sample Input :**  
2014-04-28    
   
**Sample Output :**

20 months before 2014-04-28 will be 2012-08-28

**Exercise32:** **IO - Simple File Read**

Write a program to read the given file content line by line and display to the user.

**Input File Content:**

Virat Kohli,Royal Challengers Bangalore,16

MS Dhoni,Chennai Super Kings,54

**Note:**

Create a main class "Main.java"

* Read Input filename from the user.
* Use FileReader class to read the content from the file.
* Display the file content to user.

**Input and Output Format:**

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

**Sample Input and Output:**

Enter the file name

player.txt

Virat Kohli,Royal Challengers Bangalore,16

MS Dhoni,Chennai Super Kings,54   
  
**Note:**Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

**Exercise33**: **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”.

Input and Output Format:

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

 NOTE: Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.

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

**JDBC**

**Exercise1:** **Insert Employee details using JDBC**

Write a program to insert the employee details to the database using JDBC.  
  
Perform the following instructions: 

1. Create payroll schema with table employee using the below scripts:

|  |
| --- |
| CREATE DATABASE payroll; USE payroll; CREATE TABLE employee (     employee\_id    int AUTO\_INCREMENT,     name           varchar(255),     date\_of\_birth  date,     salary         int,     PRIMARY KEY (employee\_id)); |

1. Create Employee class with attributes listed below. Include appropriate constructor and getter / setter methods.
   * employeeId
   * name
   * dateOfBirth
   * salary
2. Create EmployeeDAO to handle all the database related functions.

* Include a method getConnection() to get the connection details and connect to the database.
* Include a method addEmployee(Employee employee) to connect to the database and insert employee details to the database using prepared statement. Read the values from employee and bind the parameters.

1. Create a class EmployeeService

* Implement method addEmployee(employee), which creates an instance of EmployeeDAO class and invokes the addEmployee() method to save the data.

1. Create a class named Main and implement the following steps in main method:

* Read the detail for one employee from the console and set it in a new instance of Employee.
* Create an instance of EmployeeService and invoke addEmployee() method passing the employee instance created in the previous step.

**NOTE:**Code submission not required. To be implemented using eclipse IDE and manually verified by the SME.  
  
**Sample Input and Output:**   
   
Enter the Employee details  
Enter the ID:   
**969143**  
Enter the Name:   
**John**  
Enter the Date of birth (dd/MM/yyyy):   
**08/07/1987**  
Enter the Salary:   
**120000**  
The Given data is successfully inserted to the database.

**Exercise2:** Display Employee details using JDBC

Write a program to display the list of employees from the database using JDBC.

Perform the following instructions:

1. Modify class EmployeeDAO that was implemented as part for the previous problem.

* Include a method getAllEmployees() which returns the ArrayList of employee objects.
* Set each row retrieved from the database to a new employee instance and add it to the employeeList and return the same

2. Modify class EmployeeService to include a new method getAllEmployees() that invokes the respective method DAO class and returns the same.

3. Modify the existing main method Main class so that after insert of employee details, let is display all the rows from the database by invoking the getAllEmployees() method of EmployeeService. Use lambda expression to display the list of employees.

Refer sample output for formatting specifications.

Sample Output:

Enter the Employee details

Enter the ID:

969143

Enter the Name:

John

Enter the Date of birth (dd/MM/yyyy):

08/07/1987

Enter the Salary:

120000

Employee details inserted successfully. Refer employee list below.

John 08/07/1987 120000

Priya 24/08/1987 140000

Divya 14/08/1987 160000