1. AirVoice - Registration

Grade settings: Maximum grade: 100
Disable external file upload, paste and drop external content: Yes
Run: Yes **Evaluate**: Yes
Automatic grade: Yes **Maximum execution time**: 16 s
SmartBuy is a leading mobile shop in the town. After buying a product, the customer needs to provide a few personal details for the invoice to be generated.

You being their software consultant have been approached to develop software to retrieve the personal details of the customers, which will help them to generate the invoice faster.

Component Specification: Customer

Type(Class)	Attributes	Methods	Responsibilities
Customer	String customerName long contactNumber String emailId int age	Include the getters and setters method for all the attributes.	

In the **Main** class, create an object for the Customer class.

Get the details as shown in the sample input and assign the value for its attributes using the setters.

Display the details as shown in the sample output using the getters method.

All classes and methods should be public, Attributes should be private.

Note:

In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the rest of the text represents the output.

Ensure to follow the object oriented specifications provided in the question.

Ensure to provide the names for classes, attributes and methods as specified in the question.

Adhere to the code template, if provided.

Sample Input 1:

Enter the Name:

john

Enter the ContactNumber:

9874561230

Enter the EmailId:

john@gmail.com

Enter the Age:

32

Sample Output 1:

Name:john

ContactNumber:9874561230

EmailId:john@gmail.com

Age:32

Automatic evaluation[+]

Customer.java

```
public class Customer {
   private String customerName;

   private long contactNumber;

   private String emailId;

   private int age;

    public String getCustomerName() {
       return  customerName;
   }

    public void setCustomerName(String customerName) {
       this.customerName = customerName;
   }

    public long getContactNumber() {
       return contactNumber;
   }

    public void setContactNumber(long contactNumber) {
       this.contactNumber = contactNumber;
   }

    public String getEmailId() {
       return emailId;
   }
```

```
29
30    public void setEmailId(String emailId) {
31        this.emailId= emailId;
32    }
33
34    public int getAge() {
35        return age;
36    }
37
38    public void setAge(int age){
39        this.age = age;
40    }
41
42
43
44 }
```

Main.java

```
1  import java.util.Scanner;
2
3  public class Main {
4
5     public static void main (String[] args) {
6        Scanner sc=new Scanner(System.in);
7
8        //Fill the code
9        Customer c=new Customer();
10        System.out.println("Enter the Name:");
11        String name=(sc.nextLine());
12        System.out.println("Enter the ContactNumber:");
13        long no=sc.nextLong();
14        sc.nextLine();
15        System.out.println("Enter the EmailId:");
16        String mail=sc.nextLine();
17
18        System.out.println("Enter the Age:");
19        int age=sc.nextInt();
20        c.setCustomerName(name);
21        c.setContactNumber(no);
22        c.setEmailId(mail);
23        c.setAge(age);
24        System.out.println("Name:"+c.getCustomerName());
25        System.out.println("ContactNumber:"+c.getContactNumber());
26        System.out.println("EmailId:"+c.getEmailId());
27        System.out.println("Age:"+c.getAge());
28
29
30
31    }
32
33 }
```

Grade

Reviewed on Friday, 10 December 2021, 6:14 PM by Automatic grade

Grade 100 / 100

Assessment report

[+]**Grading and Feedback**

2. ZeeZee bank

Grade settings: Maximum grade: 100
Disable external file upload, paste and drop external content: Yes
Run: Yes **Evaluate**: Yes
Automatic grade: Yes **Maximum execution time**: 16 s
ZeeZee is a leading private sector bank. In the last Annual meeting, they decided to give their customer a 24/7 banking facility. As an initiative, the bank outlined to develop a stand-alone device that would offer deposit and withdrawal of money to the customers anytime.

You being their software consultant have been approached to develop software to implement the functionality of deposit and withdrawal anytime.

Component Specification: Account

Type(Class)	Attributes	Methods	Responsibilities
Account	long accountNumber double balanceAmount	Include the getters and setters method for all the attributes. Include a parametrized constructor of two arguments in the order – accountNumber,balanceAmount to intialize the values for the account object	

Requirement 1: Being able to deposit money into an account anytime

As per this requirement, the customer should be able to deposit money into his account at any time and the deposited amount should reflect in his account balance.

Component Specification: Account

Component Name	Type(Class)	Methods	Responsibilities
Deposit amount to an account	Account	public void deposit(double depositAmt)	This method takes the amount to be deposited as an argument This method should perform the deposit,by adding the deposited

			amount to the balanceAmount

Requirement 2: Being able to withdraw money from the account anytime

As per this requirement, the customer should be able to withdraw money from his account anytime he wants. The amount to be withdrawn should be less than or equal to the balance in the account. After the withdrawal, the account should reflect the balance amount

Component Specification: Account

Component Name	Type(Class)	Methods	Responsibilities
Withdraw amount from an account	Account	public boolean withdraw(double withdrawAmt)	This method should take the amount to be withdrawn as an argument. This method should check the balanceAmount and deduct the withdraw amount from the balanceAmount and return true. If there is insufficient balance then return false.

In the **Main** class, Get the details as shown in the sample input.

Create an object for the Account class and invoke the deposit method to deposit the amount and withdraw method to withdraw the amount from the account.

All classes and methods should be public, Attributes should be private.

Note:

Balance amount should be displayed corrected to 2 decimal places.

Order of the transactions to be performed (Display,Deposit,Withdraw).

If the balance amount is insufficient then display the message as shown in the Sample Input / Output.

In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the rest of the text represents the output.

Ensure to follow the object-oriented specifications provided in the question.

Ensure to provide the names for classes, attributes, and methods as specified in the question.

Adhere to the code template, if provided.

Sample Input/Output 1:

Enter the account number:

1234567890

Enter the available amount in the account:

15000

Enter the amount to be deposited:

1500

Available balance is:16500.00

Enter the amount to be withdrawn:

500

Available balance is:16000.00

Sample Input/Output 2:

Enter the account number:

1234567890

Enter the available amount in the account:

15000

Enter the amount to be deposited:

1500

Available balance is:16500.00

Enter the amount to be withdrawn:

18500

Insufficient balance

Available balance is:16500.00

Automatic evaluation[+]

Main.java

```java
import java.util.Scanner;
import java.text.DecimalFormat;

public class Main{

    public static void main (String[] args) {
        Scanner sc=new Scanner(System.in);
        DecimalFormat decimalFormat=new DecimalFormat("0.00");
        System.out.println("Enter the account number:");
         long accountNumber= sc.nextLong();
         System.out.println("Enter the available amount in the account:");
         double balanceAmount= sc.nextDouble();
         Account account=new Account(accountNumber,balanceAmount);
         System.out.println("Enter the amount to be deposited:");
         double depositAmount=sc.nextDouble();
         account.deposit(depositAmount);
         double availableBalance=account.getBalanceAmount();
         System.out.println("Available balance is:"+decimalFormat.format(availableBalance));
         System.out.println("Enter the amount to be withdrawn:");
         double withdrawAmount= sc.nextDouble();
         boolean isWithdrawn=account.withdraw(withdrawAmount);
         availableBalance=account.getBalanceAmount();
         if(!isWithdrawn){
             System.out.println("Insufficient balance");
         }
         System.out.println("Available balance is:"+decimalFormat.format(availableBalance));

         //Fill the code
    }
}
```

Account.java

```java

public class Account {
    private long accountNumber;
    private double balanceAmount;
    public Account(long accountNumber,double balanceAmount){
        this.accountNumber=accountNumber;
        this.balanceAmount=balanceAmount;
    }
    public long getAccountNumber(){
         return accountNumber;
    }
     public void setAccountNumber(long accountNumber){
         this.accountNumber=accountNumber;
    }
     public double getBalanceAmount(){
         return balanceAmount;
    }
     public void setBalanceAmount(double balanceAmount){
         this.balanceAmount=balanceAmount;
    }
     public void deposit(double depositAmount){
         balanceAmount+=depositAmount;
    }
```

```
24    public boolean withdraw(double withdrawAmount){
25        if(withdrawAmount<=balanceAmount){
26            balanceAmount-=withdrawAmount;
27            return true;
28        }
29        return false;
30    }
31 }
```

Grade

Reviewed on Thursday, 27 May 2021, 3:28 AM by Automatic grade
Grade 100 / 100
Assessment report
[+]**Grading and Feedback**

3.Call Details

Grade settings: Maximum grade: 100
Disable external file upload, paste and drop external content: Yes
Run: Yes **Evaluate**: Yes
Automatic grade: Yes **Maximum execution time**: 16 s
AirCarrier is a leading mobile network provider. They maintain a record of all the calls made by their postpaid customers.The details are stored in a particular format [callId:calledNumber:noOfMinutes] .At the end of every month, the network provider wants to extract the information from the file and populate it to the Call object for calculating the bill.

You being their software consultant have been approached to develop software to implement the functionality of extracting the data from the given format.

Component Specification: Call

Type(Class)	Attributes	Methods	Responsibilities
Call	int callId long calledNumber float duration	Include the getters and setters method for all the attributes.	

Requirement 1: Extracting the data from the callDetails

This requirement is responsible for extracting the customer's callId, calledNumber and duration from the callDetails. After the extraction set the callId, calledNumber and duration to the call object.

Component Specification: Call

Component Name	Type(Class)	Methods	Responsibilities
Extraction from file	Call	public void parseData(String callDetails)	This method takes the callDetails as an argument This method should perform the extraction process, to set the callId, calledNumber and duration of the call object.

In the **Main** class, Get the details as shown in the sample input.

Create an object for the Call and invoke the **parseData** method to set the callId, calledNumber and duration for each customer.

Invoke the corresponding getters to display the call details as shown in the Sample Output

All classes and methods should be public, Attributes should be private.

Note:

In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the rest of the text represents the output.

Ensure to follow the object-oriented specifications provided in the question.

Ensure to provide the names for classes, attributes and methods as specified in the question.

Adhere to the code template, if provided.

Sample Input 1:

Enter the call details:

102:6547891230:2.15

Sample Output 1:

Call id:102

Called number:6547891230

Duration:2.15

==========

Automatic evaluation[+]

Main.java

```
import java.util.Scanner;

public class Main {

    public static void main (String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the call details:");
        String a=sc.nextLine();
        Call obj=new Call();
         obj.parseData(a);
         System.out.println("Call id:"+obj.getCallId());
         System.out.println("Called number:"+obj.getCalledNumber());
         System.out.println("Duration:"+obj.getDuration());
         //Fill the code

    }
}
```

Call.java

```

public class Call {
    private int callId;
    private long calledNumber;
    private float duration;
    public Call(){
    }
    public int getCallId(){
        return callId;
    }
    public long getCalledNumber(){
        return calledNumber;
    }
    public float getDuration(){
        return duration;
    }
    public void setCallId(int callId){
        this.callId=callId;
    }
    public void setCalledNumber(long calledNumber){
        this.calledNumber=calledNumber;
    }
    public void setDuration(float duration){
        this.duration=duration;
    }
    public void parseData(String callId){
        callId=Integer.parseInt(callId.split(":")[0]);
        setCallId(callId);
        calledNumber=Long.parseLong(callId.split(":")[1]);
        setCalledNumber(calledNumber);
        duration=Float.parseFloat(callId.split(":")[2]);
        setDuration(duration);
    }
}
```

Grade

Reviewed on Tuesday, 4 May 2021, 4:58 AM by Automatic grade
Grade 100 / 100
Assessment report
[+]**Grading and Feedback**

Pair of Two digits

Grade settings: Maximum grade: 100
Disable external file upload, paste and drop external content: Yes
Run: Yes **Evaluate**: Yes
Automatic grade: Yes **Maximum execution time**: 32 s
Jerold teacher assigned a task to him. The task is to find the pair of two-digit numbers.

The pair is found by checking whether the product of the numbers is same as the product of the reversed numbers. If it is same, then print "Correct pair found". If not print, "Correct pair not found".

Write a Java program to find the correct pair of two-digit numbers.

Assume both the inputs are 2-digit values.

Note:

In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the remaining text represents the output.

Adhere to the code template, if provided.

Hint: [13*62=31*26]

Sample Input 1:

13

62

Sample Output 1:

13 and 62 are correct pair

Sample Input 2:

10

56

Sample Output 2:

10 and 56 are not correct pair

Automatic evaluation[+]

Main.java

```
1 import java.util.*;
2
3 public class Main{
4
5   public static void main (String[] args) {
6     Scanner sc=new Scanner(System.in);
7
8        int a=sc.nextInt();
9        int b=sc.nextInt();
10        if(a>99||a<10||b>99||b<10){
11            System.out.println("No");
12
13        }
14        Main obj=new Main();
15        int ra=obj.rvs(a);
16        int rb=obj.rvs(b);
17        if(a*b==ra*rb){
18            System.out.println(a+" and "+b+" are correct pair");
19        }
20        else{
21            System.out.println(a+" and "+b+" are not correct pair");
22        }
23   }
24    int rvs(int num){
25        int r,rnum=0;
26        while(num>0)
27        {
28            r=num%10;
29            rnum=rnum*10+r;
30            num/=10;
31        }
32        return(rnum);
33    }
34 }
35
36
37
38
39
40
```

Grade

Reviewed on Thursday, 27 May 2021, 3:38 AM by Automatic grade
Grade 100 / 100
Assessment report

TEST CASE PASSED
[+]**Grading and Feedback**

----------End----------

Numerology number

Grade settings: Maximum grade: 100
Disable external file upload, paste and drop external content: Yes
Run: Yes **Evaluate**: Yes
Automatic grade: Yes **Maximum execution time**: 32 s

Harry is very much interested in learning numerology with a programming Language. Help Harry to implement this task.

Write a java program to find the sum of the digits and the numerology number(Multi-digit numbers are added and reduced to a single digit), followed by the total number of odd numbers and the total number of even numbers. Assume input is greater than zero and less than 10000000.

For example, if the given number is 7654 then,

Sum of digits: 22 (7+6+5+4)

Numerology number: 4 ((7+6+5+4 =22 => 2+2) sum of digits is again added and reduced to a single digit).

Number of odd numbers: 2

Number of even numbers: 2

If the given number has 0 in it. say 80664

Sum of digits:24(8+0+6+6+4)

Numerology number:6 (2+4)

Number of odd numbers:0

Number of even numbers:5.

Note:

In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the remaining text represents the output.

Adhere to the code template, if provided.

Sample input:

Enter the number

86347

Sample output:

Sum of digits

28

Numerology number

1

Number of odd numbers

2

Number of even numbers

3

Explanation: Sum of digits=28,Numerology number: 28=>2+8=10=>1+0=1. Here, 1 is the Numerology number.

Prime numbers ending with one

Grade settings: Maximum grade: 100
Disable external file upload, paste and drop external content: Yes
Run: Yes **Evaluate**: Yes
Automatic grade: Yes **Maximum execution time**: 60 s
James is teaching about prime number to his friend John. James now assigned a task to John.

He needs to find the prime numbers that ends with one, between n1 and n2. The n1 and n2 are the two given inputs, where n1 is the lower range and n2 is the higher range. If n2 itself is prime, then stop the series. Else print the nearest prime number that ends with one, next to n2.

Write a java program to implement the above task.

Note:

In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the remaining text represents the output.

Adhere to the code template, if provided.

Sample input:

Enter the first number

10

Enter the last number

100

Sample output:

11,31,41,61,71,101

Alliteration

Grade settings: Maximum grade: 100
Disable external file upload, paste and drop external content: Yes
Run: Yes **Evaluate**: Yes
Automatic grade: Yes **Maximum execution time**: 32 s
Tina and Louis played a game where one has to tell a letter and the other has to form a sentence with all the words starting with the same letter(Case Insensitive).

The sentence should contain a minimum of 3 words and maximum any number of words. For a minimum of three words, a score of 2 is given. Each additional word gets a score of 2.

Develop a Java application to check the rules of the game and print the score as given in the sample input, output.

Note:

In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the remaining text represents the output.

Adhere to the code template, if provided.

Sample Input 1

Enter the letter

S

She sells seashells

Sample output 1

Good,You get a score of 2

Sample Input 2

Enter the letter

F

Friendly fire fighting frogs fence for fabulous French food.

Sample output 2

Good,You get a score of 14

Sample Input 3

Enter the letter

H

Hannah's home has heater

Sample output 3

Good,You get a score of 4

Sample Input 4

Enter the letter

j

Jack and Jill

Sample output 4

No score

Substitution Cipher Technique

Grade settings: Maximum grade: 100
Disable external file upload, paste and drop external content: Yes
Run: Yes **Evaluate**: Yes
Automatic grade: Yes **Maximum execution time**: 32 s
Ava sitting in the first row wants to send a secret message to her friend Mia who is sitting in the last row. She wrote a secret message in a piece of paper and passed it through her classmates. Ava has used a substitution cipher technique where every letter is replaced with the 7th alphabet before the letter in the alphabet series. Since Mia already knows the technique she easily got the exact message.

The encrypted text (input) may have numbers or special characters along with letters. If so, ignore those and convert only the letters. If space occurs between the words of input, it must occur in output also. If no letters, then there is “No hidden message”.

Develop a java application that accepts the secret message to decrypt and print the actual message to the screen. Input consists of the encrypted text.

Note:

In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the remaining text represents the output.

Adhere to the code template, if provided.

Sample Input 1

Enter the encrypted text:

Pukph

Sample Output 1

Decrypted text:

India

Sample Input 2

Enter the encrypted text:

Z23hcl @d$3#haly

Sample Output 2

Decrypted text:

Save water

Sample Input 3

Enter the encrypted text:

@67$89##^^

Sample Output 3

No hidden message

Alternate Numbers Difference

Grade settings: Maximum grade: 100
Disable external file upload, paste and drop external content: Yes
Run: Yes **Evaluate**: Yes
Automatic grade: Yes **Maximum execution time**: 32 s
Sheela is crazy on mathematics. Since decided to play a game by using a set of numbers.

The game is to find the difference between the alternate numbers in the set of numbers and find the index position of the smallest number with the largest difference. If more than one pair has the same largest difference consider the first occurrence. If the size of a set is less than 5 and greater than 10, Display “Invalid array size”.

Write a java program to implement the above task

Note:

In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the remaining text represents the output.

Adhere to the code template, if provided.

Sample Input 1:

Enter the array size

6

Enter the array elements

4

3

2

10

8

6

Sample Output 1:

1

Explanation:

The absolute value of 4-2=2, 3-10=7, 2-8=6, 10-6=4. 7 is the largest value from the difference. Then compare 3 and 10 and find the smallest value. 3 is the smallest value and index position of 3 is 1. So, output is 1

Sample Input 2:

Enter the array size

7

Enter the array elements

7

6

2

2

3

1

8

Sample Output 2:

2

Sample Input 3:

Enter the array size

15

Sample Output 3:

Invalid array size

BonBon Publishing Company

Grade settings: Maximum grade: 100
Disable external file upload, paste and drop external content: Yes
Run: Yes **Evaluate**: Yes
Automatic grade: Yes **Maximum execution time**: 16 s

Bon-Bon Publishing

Bon-Bon Publishing is an advertising company that does designing and posting of advertisements for its clients on their respective websites. Due to the festival season, many new companies have approached them for creating advertisements and requested for quotations.

So, Bon Bon plans to automate the business process for their convenience. Help them by creating a Java application to cater to their client's requirements.

To start with, the company needs to automate the calculation of the advertisement charges.

Requirements 1: Calculation of Advertisement Charges

The application needs to calculate the advertisement charges by getting the information like advertisement Id, priority, the number of days, client name, advertisement type, and data specific to the advertisement type and the base cost from the user.

Component Specification: Advertisement Class (Parent Class)

Component Name	Type(Class)	Attributes	Methods	Responsibilities
Calculation of Advertisement Charges	Advertisement	int advertisementId String priority int noOfDays String clientName	Include a public getter and setter method for all the attributes. Include a public 4 argument constructor in the given order - advertisementId, priority, noOfDays, clientName	
Calculation of Advertisement Charges	Advertisement		public float calculateAdvertisementCharge(float baseCost)	This method should be an abstract method to calculate the advertisement charges based on the base cost of the advertisement

				and return the same.

Note:

- The attributes of Advertisement class should be protected.

Component Specification: VideoAdvertisement class (Needs to be a child of Advertisement class)

Component Name	Type(Class)	Attributes	Methods	Responsibilities
Calculation of Advertisement Charges	VideoAdvertisement	int duration	Include a public getter and setter method for the attribute. Include a public 5 argument constructor in the given order - advertisementId, priority, noOfDays, clientName, duration	

Component Specification: ImageAdvertisement Class (Needs to be a child of Advertisement class)

Component Name	Type(Class)	Attributes	Methods	Responsibilities
Calculation of Advertisement Charges	ImageAdvertisement	int inches	Include a public getter and setter method for the attribute. Include a public 5 argument constructor in the given order - advertisementId, priority, noOfDays, clientName, inches	

Component Specification: TextAdvertisement Class (Needs to be a child of Advertisement class)

Component Name	Type(Class)	Attributes	Methods	Responsibilities

Calculation of Advertisement Charges	TextAdvertisement	int noOfCharacters	Include a public getter and setter method for the attribute. Include a public 5 argument constructor in the given order - advertisementId, priority, noOfDays, clientName, noOfCharacters	

Note:

- The classes VideoAdvertisement, ImageAdvertisement and TextAdvertisement should be concrete classes with private attributes, public methods and should override the calculateAdvertisementCharge method.

Advertisement Charge needs to be calculated as shown below:

Advertisement Charge = baseAdvertisementCost + Booster Cost + Service Cost

The baseAdvertisementCost is calculated based on the advertisement type as follows,

- For VideoAdvertisement :

  baseAdvertisementCost = baseCost * duration * noOfDays

- For ImageAdvertisement :

  baseAdvertisementCost = baseCost * inches * noOfDays

- For TextAdvertisement :

  baseAdvertisementCost = baseCost * noOfCharacters * noOfDays

The booster cost and service cost is calculated based on the following criteria.

Priority	Booster Cost	Service Cost
high	10% of baseAdvertisementCost	1000
medium	7% of baseAdvertisementCost	700
low	Nil	200

Example:

Consider an advertisement type as video, priority as high, number of days as 5, duration as 10 minutes and base cost as 100, then the calculation of advertisement charge will be as follows,

baseAdvertisementCost = 100 * 10 * 5 = 5000

Booster Cost = (5000 * 10) / 100 = 500

Service Cost = 1000

Advertisement Charge = 5000 + 500 + 1000 = 6500.0

Create a **public class Main** with the main method to test the application.

Note:

- In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the rest of the text represents the output.
- Ensure to follow object oriented specifications provided in the question.
- Ensure to provide the names for classes, attributes and methods as specified in the question.
- Adhere to the code template, if provided.

Sample Input 1:

Enter the advertisement id

101

Enter the priority (high / medium / low)

high
Enter the no of days advertisement is published

10
Enter the client name

EZ Company
Enter the type of Advertisement (video/image/text)

video
Enter the duration in minutes

3
Enter the base cost

100

Sample Output 1:

The Advertisement cost is 4300.0

Sample Input 2:

Enter the advertisement id

247
Enter the priority (high / medium / low)

medium

Enter the no of days advertisement is published

15
Enter the client name

Zee Technologies
Enter the type of Advertisement (video/image/text)

image
Enter the number of inches

30
Enter the base cost

5

Sample Output 2:

The Advertisement cost is 3107.5

Sample Input 3:

Enter the advertisement id

355
Enter the priority (high / medium / low)

low
Enter the no of days advertisement is published

10
Enter the client name

Airvoice communications
Enter the type of Advertisement (video/image/text)

text
Enter the number of characters

100
Enter the base cost

2

Sample Output 3:

The Advertisement cost is 2200.0

Bank Account - Interface

Grade settings: Maximum grade: 100
Disable external file upload, paste and drop external content: Yes
Run: Yes **Evaluate**: Yes
Automatic grade: Yes **Maximum execution time**: 16 s

Bank Account

KGFB is a financial institution that provides facilities such as current account and savings account to its customers. To add to its facilities they have come up with the locker that is chargeable. Since it is a new facility introduced, many customers have approached them to enquire about this facility.

To make the banking service effective, KGFB plans to automate the calculation of the locker maintenance charges based on the various account types and the duration.(in years).

Help them by creating a Java application that caters to their requirement.

Requirement 1: Maintenance Charge Calculation

The customer needs to provide information like account type, customer name, account number, balance and number of years to calculate the maintenance charge.

Component Specification: Account Class (Parent Class)

Type (Class)	Attributes	Methods	Responsibilities
Account	String accountNumber String customerName double balance	Include a public getter and setter methods. Include a public 3 argument constructor in the order - accountNumber, customerName, balance	

Component Specification: MaintenanceCharge (Interface)

Component Name	Type (Interface)	Methods	Responsibilities
Maintenance Charge Calculation	MaintenanceCharge	public float calculateMaintenanceCharge (float noOfYears)	This is an abstract method and it should calculate the maintenance charge based on the number of years and account type and return the result.

Component Specification: CurrentAccount Class (should be a child of Account Class and implement MaintenanceCharge interface)

Component Name	Type(Class)	Methods	Responsibilities
Maintenance Charge Calculation	CurrentAccount	Include a public 3 argument constructor in the order - customerName, accountNumber, balance	

Component Specification: SavingsAccount Class (should be a child of Account Class and implement MaintenanceCharge interface)

Component Name	Type(Class)	Methods	Responsibilities
Maintenance Charge Calculation	SavingsAccount	Include a public 3 argument constructor in the order - customerName, accountNumber, balance	

Both the child classes CurrentAccount and SavingsAccount should override the calculateMaintenanceCharge(float noOfYears) method.

The calculateMaintenanceCharge(float noOfYears) should return the maintenance charge based on the number of years and the account type.

The Maintenance Charge needs to be calculated as shown below:

For Savings Account: Maintenance Charge = (m * n) + 50

For Current Account: Maintenance Charge = (m * n) + 200

Here, m is the base charge per year and n is the number of years. The base charge m, is Rs.50 for a Savings account and Rs.100 for a Current account. The number of years n, should be taken as input from the user as noOfYears.

Implement the calculateMaintenanceCharge method in the CurrentAccount and the SavingsAccount class based on the above assumption.

All class, interface and methods need to be public and attributes need to be private.

Write a class UserInterface with the main method to test the above functionality.

Note:

· In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the rest of the text represents the output.

· Ensure to follow the object oriented specifications provided in the question.

· Ensure to provide the names for classes, interface, attributes and methods as specified in the question.

· Adhere to the code template, if provided.

Sample Input 1:

1. Savings Account

2. Current Account

Enter your choice:

2

Enter the Account number

123456789

Enter the Customer Name

Rohit

Enter the Balance amount

5000

Enter the number of years

2

Sample Output 1:

Customer Name Rohit

Account Number 123456789

Account Balance 5000.0

Maintenance Charge for Current Account is Rs 400.0

Sample Input 2:

1. Current Account

2. Savings Account

Enter your choice:

1

Enter the Account number

123456798

Enter the Customer Name

Rohini

Enter the Balance amount

5000

Enter the number of years

2

Sample Output 2:

Customer Name Rohini

Account Number 123456798

Account Balance 5000.0

Maintenance Charge for Savings Account is Rs 150.0

Payment - Inheritance

Grade settings: Maximum grade: 100
Disable external file upload, paste and drop external content: Yes
Run: Yes **Evaluate**: Yes
Automatic grade: Yes **Maximum execution time**: 16 s

Payment Status

Roy is a wholesale cloth dealer who sells cloth material to the local tailors on monthly installments. At the end of each month, he collects the installment amount from all his customers. Some of his customers pay by Cheque, some pay by Cash and some by Credit Card. He wants to automate this payment process.

Help him to do this by writing a java program.

Requirement 1: Make Payment

The application needs to verify the payment process and display the status report of payment by getting the inputs like due amount, payment mode and data specific to the payment mode from the user and calculate the balance amount.

Component Specification: Payment Class (Parent Class)

Component Name	Type(Class)	Attributes	Methods	Responsibilities
Make payment for EMI amount	Payment	int dueAmount	Include a public getter and setter method	
Make payment for EMI amount	Payment		public boolean payAmount()	The boolean payAmount() method should return true if there is no due to be paid, else return false.

Note:

· The attributes of Payment class should be private.

· The payment can be of three types: Cheque, Cash, Credit Card.

Component Specification: Cheque class (Needs to be a child of Payment class)

Component Name	Type(Class)	Attributes	Methods	Responsibilities
	Cheque	String chequeNo int chequeAmount Date dateOfIssue	Include a public getter and setter method for all the attributes.	
Make payment for EMI amount	Cheque		public boolean payAmount()	This is an overridden method of the parent class. It should return true if the cheque is valid and the amount is valid. Else return false.

Note:

- The cheque is valid for 6 months from the date of issue.
- Assume the current date is 01-01-2020 in dd-MM-yyyy format.
- The chequeAmount is valid if it is greater than or equal to the dueAmount.

Component Specification: Cash class (Needs to be a child of Payment class)

Component Name	Type(Class)	Attributes	Methods	Responsibilities
Make payment for EMI amount	Cash	int cashAmount	Include a public getter and setter method for the attribute.	
Make payment for EMI amount	Cash		public boolean payAmount()	This is an overridden method of the parent class. It should return true if the cashAmount is greater than or equal to the dueAmount. Else return false.

Component Specification: Credit class (Needs to be a child of Payment class)

Component Name	Type (Class)	Attributes	Methods	Responsibilities
Make payment for EMI amount	Credit	int creditCardNo	Include a public getter and setter	

		String cardType int creditCardAmount	method for all the attributes.	
Make payment for EMI amount	Credit		public boolean payAmount()	This is an overridden method of the parent class. It should deduct the dueAmount and service tax from the creditCardAmount and return true if the credit card payment was done successfully. Else return false.

Note:

· The payment can be done if the credit card amount is greater than or equal to the sum of due amount and service tax. Else payment cannot be made.

· The cardType can be “silver” or “gold” or “platinum”. Set the creditCardAmount based on the cardType.

· Also service tax is calculated on dueAmount based on cardType.

Credit Card Type	Credit Card Amount	Service Tax
silver	10000	2% of the due amount
gold	50000	5% of the due amount
platinum	100000	10% of the due amount

· The boolean payAmount() method should deduct the due amount and the service tax amount from a credit card. If the creditCardAmount is less than the dueAmount+serviceTax, then the payment cannot be made.

· The balance in credit card amount after a successful payment should be updated in the creditCardAmount by deducting the sum of dueAmount and serviceTax from creditCardAmount itself.

Component Specification: Bill class

Component Name	Type(Class)	Attributes	Methods	Responsibilities

Payment Status Report	Bill		public String processPayment (Payment obj)	This method should return a message based on the status of the payment made.

Note:

· If the payment is successful, processPayment method should return a message “Payment done successfully via cash” or “Payment done successfully via cheque” or “Payment done successfully via creditcard. Remaining amount in your <<cardType>> card is <<balance in CreditCardAmount>>”

· If the payment is a failure, then return a message “Payment not done and your due amount is <<dueAmount>>”

Create a **public class Main** with the main method to test the application.

Note:

· Assume the current date as 01-01-2020 in dd-MM-yyyy format.

· In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the rest of the text represents the output.

· Ensure to follow the object oriented specifications provided in the question.

· Ensure to provide the names for classes, attributes and methods as specified in the question.

· Adhere to the code template, if provided.

· Adhere to the sample input and output.

Sample Input 1:

Enter the due amount:

3000
Enter the mode of payment(cheque/cash/credit):

cash
Enter the cash amount:

2000

Sample Output 1:

Payment not done and your due amount is 3000

Sample Input 2:

Enter the due amount:

3000
Enter the mode of payment(cheque/cash/credit):

cash
Enter the cash amount:

3000

Sample Output 2:

Payment done successfully via cash

Sample Input 3:

Enter the due amount:

3000
Enter the mode of payment(cheque/cash/credit):

cheque
Enter the cheque number:

123
Enter the cheque amount:

3000
Enter the date of issue:

21-08-2019

Sample Output 3:

Payment done successfully via cheque

Sample Input 4:

Enter the due amount:

3000
Enter the mode of payment(cheque/cash/credit):

credit
Enter the credit card number:

234
Enter the card type(silver,gold,platinum):

silver

Sample Output 4:

Payment done successfully via credit card. Remaining amount in your silver card is 6940

GPA Calculation

Grade settings: Maximum grade: 100
Disable external file upload, paste and drop external content: Yes
Run: Yes **Evaluate**: Yes
Automatic grade: Yes **Maximum execution time**: 16 s

Ranveer is an engineering student. He wants to calculate his GPA (Grade Point Average) scored in his current semester.

To calculate the GPA, first store the details of the grade scored in all the subjects in a semester and then calculate the GPA based on the grade point.

As a java programmer, help him to do this by writing a java program to develop a GPA calculator.

Component Specification: GPACalculator Class

Type(Class)	Attributes	Methods	Responsibilities
GPACalculator	List<Integer> gradePointList	Include the getter and setter methods for the attribute.	

Requirement 1: Store the grade points scored by a student

When Ranveer gets his semester results, store the grade points scored by him for calculating the GPA.

The addGradePoint method accepts grades obtained as an argument. This method should convert the grade to the appropriate grade point and add it to the gradePointList.

Grade point is set based on the grade obtained as shown below.

Grade	S	A	B	C	D	E

Grade Point	10	9	8	7	6	5

Grade is case sensitive.

Component Specification: GPACalculator Class

Component Name	Type (Class)	Methods	Responsibilities
Add the grade points obtained by the student to the gradePointList	GPACalculator	public void addGradePoint (char gradeObtained)	This method takes the grade as an argument, converts the grade to the grade point and adds it to the gradePointList.

Requirement 2: Calculate the GPA

Ranveer needs to know his performance level for further up gradation. To know this, the GPA needs to be calculated.

The method calculateGPAScored should calculate the GPA based on the values populated in the gradePointList.

Component Specification: GPACalculator Class

Component Name	**Type(Class)**	**Methods**	**Responsibilities**
Fetch the grade points available in the list and calculate the GPA.	GPACalculator	public double calculateGPAScored ()	This method needs to calculate the GPA based on the values in the gradePointList and return the result. If the gradePointList is empty, then the method should return 0.

The GPA needs to be calculated as:

GPA= (gradePoint1 + gradePoint2 + ... + gradePointN) / (size of List)

Create a Main class with the main method.

Design the menu as described in the Sample Input and Output as:

1. Add Grade

2. Calculate GPA

3. Exit

Enter your choice

When the choice is 1, get the grade from the user and add it to the gradePointList.

When the choice is 2, display the GPA scored.

The entire program should be executed within a loop. It should not terminate after the completion of a functionality. When the choice provided is 3, it should terminate with a message "Thank you for using the Application".

Please do not use System.exit(0). Instead use a break to terminate the program.

Note:

In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the rest of the text represents the output.

Ensure to follow the object oriented specifications provided in the question description.
Ensure to provide the names for the classes, attributes and methods as specified in the question description.

Adhere to the code template, if provided.

Sample Input / Output 1:

1. Add Grade

2. Calculate GPA

3. Exit

Enter your choice

1

Enter the obtained grade

S

1. Add Grade

2. Calculate GPA

3. Exit

Enter your choice

1

Enter the obtained grade

A

1. Add Grade

2. Calculate GPA

3. Exit

Enter your choice

1

Enter the obtained grade

B

1. Add Grade

2. Calculate GPA

3. Exit

Enter your choice

2

GPA Scored

9.0

1. Add Grade

2. Calculate GPA

3. Exit

Enter your choice

3

Thank you for using the Application

Sample Input / Output 2:

1. Add Grade

2. Calculate GPA

3. Exit

Enter your choice

2

No GradePoints available

1. Add Grade

2. Calculate GPA

3. Exit

Enter your choice

3

Thank you for using the Application

Batting Average

Grade settings: Maximum grade: 100
Disable external file upload, paste and drop external content: Yes
Run: Yes **Evaluate**: Yes
Automatic grade: Yes **Maximum execution time**: 16 s

National Cricket Academy (NCA) wants to monitor the performance of the players registered with them based on the average runs scored in the matches played.

As a software consultant of NCA, help them by writing a java program to meet their requirements.

The application needs to store the runs scored by a player and calculate the average score.

Component Specification: Player Class

Type(Class)	Attributes	Methods	Responsibilities
Player	List<Integer> scoreList	Include the getter and setter methods for the attribute.	

Requirement 1: Store the runs scored by a player

Whenever a match is over, NCA needs to store the runs scored by the players for future reference.

The addScoreDetails method accepts runScored as argument and adds it to the scoreList.

Component Specification: Player Class

Component Name	Type (Class)	Methods	Responsibilities
Add runs scored to the scoreList	Player	public void addScoreDetails(int runScored)	This method takes the runScored as an argument, and adds it to the scoreList.

Requirement 2: Calculate the average run scored

NCA Academy needs to monitor the performance of a player periodically based on the average runs scored.

The method getAverageRunScored should calculate the average runs scored based on the scoreList.

Component Specification: Player Class

Component Name	Type(Class)	Methods	Responsibilities
Fetch the runs scored in each match from the list and calculate the	Player	public double getAverageRunScored()	This method needs to calculate the average runs scored based on the scoreList and return the result.

average runs scored.			If the scoreList is empty, the method should return 0.

The average runs scored needs to be calculated as:

Average runs scored = sum of all runs available in the scoreList / size of the scoreList

Create a Main class with the main method.

Design the menu as described in the Sample Input and Output as:

1. Add runs scored

2. Calculate average runs scored

3. Exit

Enter your choice

When the choice is 1, get the runs scored from the user and add it to the scoreList.

When the choice is 2, display the average runs scored.

The entire program should be executed within a loop. It should not terminate after the completion of a functionality. When the choice provided is 3, it should terminate with a message "Thank you for using the Application".

Please do not use System.exit(0). Instead use break to terminate the program.

Note:

In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the rest of the text represents the output.

Ensure to follow the object oriented specifications provided in the question description.
Ensure to provide the names for the classes, attributes and methods as specified in the question description.

Adhere to the code template, if provided.

Sample Input / Output 1:

1. Add Runs Scored

2. Calculate average runs scored

3. Exit

Enter your choice

1

Enter the runs scored

150

1. Add Runs Scored

2. Calculate average runs scored

3. Exit

Enter your choice

1

Enter the runs scored

50

1. Add Runs Scored

2. Calculate average runs scored

3. Exit

Enter your choice

1

Enter the runs scored

50

1. Add Runs Scored

2. Calculate average runs scored

3. Exit

Enter your choice

2

Average runs secured

83.33333333333333

1. Add Runs Scored

2. Calculate average runs scored

3. Exit

Enter your choice

3

Thank you for using the Application

Hunger Eats

Grade settings: Maximum grade: 100
Disable external file upload, paste and drop external content: Yes
Run: Yes **Evaluate**: Yes
Automatic grade: Yes **Maximum execution time**: 16 s

Hunger Eats wants to establish their business and promote their sales. So, they decide to give a maximum of 50% discount to their customers on the food ordered based on the bank that they choose to make the payment. [HDFC, ICICI, CUB, and SBI].

When a customer places an order for food, the order needs to be stored to the cart and the discount needs to be calculated based on the bank selected for payment. Also, the bill amount for the order needs to be calculated.

As a software developer, help them to automate the above process by writing a simple java program.

Component Specification: FoodProduct (model Class)

Type(Class)	Attributes	Methods	Responsibilities
FoodProduct	int foodId String foodName double costPerUnit int quantity	Include getters and setters for all the attributes.	

Component Specification: Order (Utility Class)

Type(Class)	Attributes	Methods	Responsibilities
Order	List<FoodProduct> foodList double discountPercentage	Include the getters and setter method for all the attributes.	

Requirement 1: Add Food Product to the list

Whenever a customer places an order, the admin has to add food products ordered by the customer to the cart.

The method addToCart takes FoodProduct object as an argument and adds it to the foodList.

Component Specification: Order (Utility Class)

Component Name	Type(Class)	Methods	Responsibilities
Add FoodProduct object to the foodList	Order	public void addToCart (FoodProduct foodProductObject)	This method takes an object of FoodProduct as an argument and adds it to the foodList.

Requirement 2: Find the discount percentage

Hunger Eats provides a discount to the customers for the orders placed. This discount is based on the bank selected by the customer for payment.

BANK	Discount Percentage
HDFC	15
ICICI	25
CUB	30
SBI	50
OTHERS	0

Assumption: All Bank Names are in Upper case as specified in the Table above

Method findDiscount takes the bank name as an argument and sets the value for the attribute discountPercentage based on the bank name.

Component Specification: Order (Utility Class)

Component Name	Type(Class)	Methods	Responsibilities

Find discount percentage based on the bank name selected by the customer	Order	public void findDiscount(String bankName)	This method takes bankName as an argument, and assigns value to the discountPercentage attribute of the Order class.

Requirement 3: Calculate the total bill

An order can have multiple food products. For each order placed, the bill amount needs to be calculated.

The bill amount is calculated as shown below:

Total Price = sum (cost * quantity) of all food products in the list.

Bill Amount =Total Price - (Total Price * discountPercentage / 100)

The method calculateTotalBill calculates the bill amount by fetching the food details from the foodList and returns the calculated amount.

Component Specification: Order (Utility Class)

Component Name	Type(Class)	Methods	Responsibilities
Calculate the total bill amount for the order placed by the customer.	Order	public double calculateTotalBill ()	This method fetches the FoodProduct objects in foodList and calculates the bill amount and returns the result If the foodList is empty this method should return 0.

Note:

In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the rest of the text represents the output.

Ensure to follow the object oriented specifications provided in the question description.

Ensure to provide the names for the classes, attributes and methods as specified in the question description.

Adhere to the code template, if provided.

Sample Input / Output 1:

Enter the number of items

3

Enter the item details

Enter the item id

1

Enter the item name

Veg fried rice

Enter the cost per unit

180

Enter the quantity

2

Enter the item id

2

Enter the item name

Mushroom Manchurian

Enter the cost per unit

130

Enter the quantity

1

Enter the item id

3

Enter the item name

Lemon soda

Enter the cost per unit

80

Enter the quantity

2

Enter the bank name to avail offer

ICICI

Calculated Bill Amount: 487.5