Question 1 : - Write a Program that simulates a telephone that records missed incoming calls. For each missed call, store the time of call, telephone number of origin, and name of the caller if the name is available. For unlisted numbers, set the name to "private caller". Choose or extend the most appropriate collection class and provide the following features.

a) Numbers are recalled in the order they arrive
b) Up to 10 numbers are recorded. When the eleventh call comes in, it is stored and the oldest call is deleted so that no more than 10 numbers are ever recorded.
c) After each number display, the user can select
    i. To delete the call
    ii. To go on to the next missed call, or
    iii. To display the call details (number, caller name and time). Delete the number if user specifies a number to delete.

Write a helper class to represent an incoming call with fields to hold the number, name of the caller, and time of the call. Write a tester call that stores the several numbers, simulate the user pressing the missed-calls button, and finally prints the entire collection of stored calls.

Solution :-

Contact.java

```
package telephone;
public class Contact {
  long phoneNumber;
  String name;

  public Contact(String name, long phoneNumber) {
    this.name = name;
    this.phoneNumber = phoneNumber;
  }

  public String toString() {
    return "Name of the customer is :- " + name + ", And his Phone Number is :- "
+ phoneNumber;
  }
}
```

ContactList.java

```
package telephone;
import java.util.*;

public class ContactList {
  Scanner sc = new Scanner(System.in);
  static ArrayList<Contact> CustomerList = new ArrayList<Contact>();

  public ContactList() {
    while (true) {
      System.out.println("Enter 1 to enter the details OR 0 to exit :- ");
      if (sc.nextLine().equals("1")) {
        System.out.println("Enter name of the customer :- ");
        String name = sc.nextLine();
        System.out.println("Enter Phone Number of the customer :- ");
```

```
        long phoneNumber = sc.nextLong();
        sc.nextLine();
        Contact newCustomerDetails = new Contact(name, phoneNumber);
        CustomerList.add(newCustomerDetails);
      }
      else
        break;
    }
  }
  public static ArrayList<Contact> getCustomerList() {
    return CustomerList;
  }
  public void viewList() {
    for (Contact itr : CustomerList)
      System.out.println(itr);
  }
}
```

Main.java

```
package telephone;
public class Main {
  public static void main(String args[]) {
    ContactList CustomerListObject = new ContactList();
    CustomerListObject.viewList();
    MissedCallList missedCallListObject = new MissedCallList();
    missedCallListObject.viewLog();
  }
}
```

MissedCall.java

```
package telephone;
import java.time.LocalDateTime;
public class MissedCall {
  String name = "private caller";
  long phone_Number;
  LocalDateTime localTime;

  public MissedCall(long phone_Number) {
    this.phone_Number = phone_Number;
    this.localTime = LocalDateTime.now();
    for (Contact itr : ContactList.getCustomerList()) {
      if (itr.phoneNumber == phone_Number) {
        this.name = itr.name;
        break;
      }
    }
  }
```

```
  public String toString() {
    return "Name of the customer is :- " + name + ", and his Phone number is :- "
+ phone_Number + ", He/she called at time :- " + localTime;
  }
}
```

MissedCallList.java

```
package telephone;
import java.util.*;
public class MissedCallList {
  Scanner sc = new Scanner(System.in);
  ArrayList<MissedCall> missedCallList = new ArrayList<MissedCall>();
  public MissedCallList() {
    while (true) {
      System.out.println("Enter yes to give missedcall :- ");
      if (sc.nextLine().equals("yes")) {
        System.out.println("Enter number : ");
        long num = Long.parseLong(sc.nextLine());
        if (missedCallList.size() > 4)
          missedCallList.remove(0);
        missedCallList.add(new MissedCall(num));
      }
      else
        break;
    }
  }

  public void viewLog() {
    Iterator<MissedCall> itr = missedCallList.iterator();
    while (true) {
      System.out.println("Enter option\n1.) delete any missedcall\n2.) display
missedcall\n3.) Exit");
      int choice = sc.nextInt();
      sc.nextLine();
      if (choice == 1) {
        System.out.println("Enter the number whose call you wish to delete: ");
        long phoneNumberToBeDeleted = Long.parseLong(sc.nextLine());
        while (itr.hasNext())
          if (itr.next().phone_Number == phoneNumberToBeDeleted)
            itr.remove();
        System.out.println("Successfully deleted");
      }
      else if (choice == 2)
        for (MissedCall it : missedCallList)
          System.out.println(it);
      else
        break;
    }
  }
}
```

Output :-

```
Enter 1 to enter the details OR 0 to exit :-
1
Enter name of the customer :-
agam
Enter Phone Number of the customer :-
123
Enter 1 to enter the details OR 0 to exit :-
1
Enter name of the customer :-
aman
Enter Phone Number of the customer :-
456
Enter 1 to enter the details OR 0 to exit :-
1
Enter name of the customer :-
amar
Enter Phone Number of the customer :-
789
Enter 1 to enter the details OR 0 to exit :-
0
Name of the customer is :- agam, And his Phone Number is :- 123
Name of the customer is :- aman, And his Phone Number is :- 456
Name of the customer is :- amar, And his Phone Number is :- 789
Enter yes to give missedcall :-
yes
Enter number :
123
Missed Call has been done on the number :- 123
Enter yes to give missedcall :-
yes
Enter number :
456
Missed Call has been done on the number :- 456
Enter yes to give missedcall :-
yes
Enter number :
789
Missed Call has been done on the number :- 789
Enter yes to give missedcall :-
no
Enter option
1.) delete any missedcall
2.) display missedcall
3.) Exit
1
Enter the number whose call you wish to delete:
123
Successfully deleted
Enter option
1.) delete any missedcall
2.) display missedcall
3.) Exit
2
Name of the customer is :- aman, and his Phone number is :- 456, He/she called at time :- 2022-04-26T00:24:36.446516200
Name of the customer is :- amar, and his Phone number is :- 789, He/she called at time :- 2022-04-26T00:24:40.966692800
Enter option
1.) delete any missedcall
2.) display missedcall
3.) Exit
3
```

Question 2 :- Write a Java program using user-defined storage classes to create a book database and store it in a Collection List.

a) Books collection should include title, author, publisher and price.
b) Write a method to sort the books in ascending order of price and store it in another List. Maintain the book details with respect to an unique book id.
c) Prompt for an author name and list all the books with the same author name. Create a new list holding all the book details with price greater than a user specified price.
d) For a given a value by the user, find all the books that match either the whole or a part of the book title.
e) Identify a publisher and print books from a particular publisher. Update the publisher details based on a title.

Solution :-

Book.java

```
package questionNo2;
public class Book {
  String title, authorName, publisherName;
  double price;

  public Book(String title, String authorName, String publisherName, double price)
  {
    this.title = title;
    this.authorName = authorName;
    this.publisherName = publisherName;
    this.price = price;
  }

  public String toString() {
    return "Title of the book is :- " + title + ", whose Author name is :- " +
authorName
        + ", and the name of the Publisher is :- " + publisherName + ", The price
of the Book is :- " + price;
  }
}
```

BookCompare.java

```
package questionNo2;
import java.util.*;

public class BookCompare implements Comparator<Book> {
  public int compare(Book book1, Book book2) {
    return (int) (book1.price - book2.price);
  }
}
```

BookList.java

```
package questionNo2;
import java.util.*;

public class BookList {
  Scanner sc = new Scanner(System.in);
  HashMap<Integer, Book> hashMap = new HashMap<>();
  LinkedList<Book> bookDetailsLL = new LinkedList<Book>();
```

```
  public void BookHashmap(ArrayList<Book> bookList) {
    int i = 1;
    for (Book itr : bookList) {
      hashMap.put(i, itr);
      i++;
    }
  }

  public void getInput(ArrayList<Book> arrayListOfBookDetails) {
    while (true) {
      System.out.println("Do you want to enter details of books? Yes or No: ");
      if (sc.nextLine().equals("Yes")) {
        System.out.println("Enter Title of the Book :- ");
        String title = sc.nextLine();
        System.out.println("Enter Author Name of the Book:- ");
        String author = sc.nextLine();
        System.out.println("Enter Publisher Name of the Book :- ");
        String publisher = sc.nextLine();
        System.out.println("Enter Price of the Book :- ");
        double price = Double.parseDouble(sc.nextLine());
        Book newBookDetails = new Book(title, author, publisher, price);
        arrayListOfBookDetails.add(newBookDetails);
      } else
        break;
    }
  }

  // printing the contents of the map
  public void printHashMap() {
    for (Map.Entry<Integer, Book> itr : hashMap.entrySet()) {
      System.out.println(itr.getKey());
      System.out.println(itr.getValue());
    }
  }

  public void searchBookByTitleName(String title, ArrayList<Book>
arrayListOfBookDetails) {
    int titleLength = title.length();
    for (Book singleBookDetails : arrayListOfBookDetails) {
      int singleBookLength = singleBookDetails.title.length();
      if (singleBookDetails.title.substring(singleBookLength - titleLength,
singleBookLength).equals(title))
        System.out.println(singleBookDetails);
    }
  }

  public void searchByAutherName(String autherName) {
    for (Map.Entry<Integer, Book> itr : hashMap.entrySet()) {
      Book singleBookObject = itr.getValue();
      String currentObjectAuthorName = singleBookObject.authorName;
```

```
      if (currentObjectAuthorName.equals(autherName))
        System.out.println(itr.getValue());
    }
  }

  public void searchByPublisherName(String publisher_Name) {
    for (Map.Entry<Integer, Book> itr : hashMap.entrySet()) {
      Book singleBookObject = itr.getValue();
      if (singleBookObject.publisherName.equals(publisher_Name))
        System.out.println(singleBookObject);
    }
  }

  public void group_price(double Price) {
    for (Map.Entry<Integer, Book> itr : hashMap.entrySet()) {
      Book singleBookObject = itr.getValue();
      if (singleBookObject.price > Price)
        bookDetailsLL.add(singleBookObject);
    }
    System.out.println("Books with price greater than " + Price);
    for (Book itr : bookDetailsLL)
      System.out.println(itr);
  }

  public void updatePublisherName(String enteredTitle, String publisher_Name) {
    for (Map.Entry<Integer, Book> itr : hashMap.entrySet()) {
      Book singleBookObject = itr.getValue();
      if (singleBookObject.title.equals(enteredTitle)) {
        singleBookObject.publisherName = publisher_Name;
        hashMap.put(itr.getKey(), singleBookObject);
      }
    }
    printHashMap();
  }
}
```

Main.java

```
package questionNo2;
import java.util.*;

public class Main {
  public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    BookList bookListObject = new BookList();
    ArrayList<Book> arrayListOfBookDetails = new ArrayList<Book>();
    bookListObject.getInput(arrayListOfBookDetails);

    // Copying the content of arrayListOfBookDetails to booklist
    ArrayList<Book> booklist = new ArrayList<Book>();
    for (Book itr : arrayListOfBookDetails)
      booklist.add(itr);
```

```
    // Sort the Book Details according to the price and print the BookList
    booklist.sort(new BookCompare());
    System.out.println("After sorting: ");
    for (Book itr : booklist)
      System.out.println(itr);

    bookListObject.BookHashmap(booklist);

    // Search Book by TITLE NAME
    System.out.println("Enter the title of book to search: ");
    String title = sc.nextLine();
    bookListObject.searchBookByTitleName(title, arrayListOfBookDetails);

    System.out.println("Enter the auther name to search his books that are present
in the BookList :- ");
    String autherName = sc.nextLine();
    bookListObject.searchByAutherName(autherName);

    System.out.println("Enter publisher name to search in bookList :- ");
    String publisherName = sc.nextLine();
    bookListObject.searchByPublisherName(publisherName);

    System.out.println("Enter title of book whose publisher you wish to update :-
");
    String enteredTitle = sc.nextLine();
    System.out.println("Enter new publisher details :- ");
    String updatePublisherName = sc.nextLine();
    bookListObject.updatePublisherName(enteredTitle, updatePublisherName);
    sc.close();
  }
}
```

Output :-

```
Do you want to enter details of books? Yes or No:
Yes
Enter Title of the Book :-
math
Enter Author Name of the Book:-
agam
Enter Publisher Name of the Book :-
s chand
Enter Price of the Book :-
123
Do you want to enter details of books? Yes or No:
Yes
Enter Title of the Book :-
physics
Enter Author Name of the Book:-
amar
Enter Publisher Name of the Book :-
Bharti Bhawan
Enter Price of the Book :-
456
Do you want to enter details of books? Yes or No:
Yes
```

```
Enter Title of the Book :-
Biology
Enter Author Name of the Book:-
Aman
Enter Publisher Name of the Book :-
NCERT
Enter Price of the Book :-
789
Do you want to enter details of books? Yes or No:
No
After sorting:
Title of the book is :- math, whose Author name is :- agam, and the name of the Publisher is :-
s chand, The price of the Book is :- 123.0
Title of the book is :- physics, whose Author name is :- amar, and the name of the Publisher is
:- Bharti Bhawan, The price of the Book is :- 456.0
Title of the book is :- Biology, whose Author name is :- Aman, and the name of the Publisher is
:- NCERT, The price of the Book is :- 789.0
Enter the title of book to search:
math
Title of the book is :- math, whose Author name is :- agam, and the name of the Publisher is :-
s chand, The price of the Book is :- 123.0
Enter the auther name to search his books that are present in the BookList :-
amar
Title of the book is :- physics, whose Author name is :- amar, and the name of the Publisher is
:- Bharti Bhawan, The price of the Book is :- 456.0
Enter publisher name to search in bookList :-
s chand
Title of the book is :- math, whose Author name is :- agam, and the name of the Publisher is :-
s chand, The price of the Book is :- 123.0
Enter title of book whose publisher you wish to update :-
Biology
Enter new publisher details :-
aman mittal
1
Title of the book is :- math, whose Author name is :- agam, and the name of the Publisher is :-
s chand, The price of the Book is :- 123.0
2
Title of the book is :- physics, whose Author name is :- amar, and the name of the Publisher is
:- Bharti Bhawan, The price of the Book is :- 456.0
3
Title of the book is :- Biology, whose Author name is :- Aman, and the name of the Publisher is
:- aman mittal publication, The price of the Book is :- 789.0
```

Question 3 :- Create a desktop java application using swings to enable an user to enter student information such as name, usn, age, address, sgpa of 8 semesters, category.

a) Perform validations on all the fields. Display appropriate messages in pop up boxes to indicate wrong entries.

b) On clicking of the “compute” button, find the cgpa . On clicking of the “done” button, mouse place the student details in a collection.

c) Display the collection in a textarea on the click of a button.

Solution :-

Student.java

```
package question3;
public class Student {
  String name, usn, address, category;
  int age;
  // float sgpa1,sgpa2,sgpa3,sgpa4,sgpa5,sgpa6,sgpa7,sgpa8;
  float cgpa;

  public Student(String name, String usn, String address, String category, int
age, float cgpa) {
    this.name = name;
    this.usn = usn;
    this.address = address;
    this.category = category;
    this.age = age;
    this.cgpa = cgpa;
  }

  public String toString() {
    return ("Name of the student is :- " + name + " with USN :- " + usn + "
residing in " + address
        + " belonging to category :- " + category + " of age :- " + age + " and
has cgpa :- " + cgpa);
  }
}
```

StudentClass.java

```
package question3;

import java.awt.*;
import java.awt.event.*;
import java.util.LinkedList;
import javax.swing.*;

public class StudentClass implements ActionListener {
  LinkedList<Student> studentList = new LinkedList<Student>();

  // Defining all the levels that are required
  JLabel Name = new JLabel("Enter Name of the Student :- ");
  JLabel USN = new JLabel("Enter USN of the Student :- ");
  JLabel Age = new JLabel("Enter Age of the Student :- ");
```

```
JLabel Address = new JLabel("Enter Address at which Student Lives :- ");
JLabel Category = new JLabel("Select category of the Student :- ");
JLabel SGPA1 = new JLabel("Enter SGPA of I sem");
JLabel SGPA2 = new JLabel("Enter SGPA of II sem");
JLabel SGPA3 = new JLabel("Enter SGPA of III sem");
JLabel SGPA4 = new JLabel("Enter SGPA of IV sem");
JLabel SGPA5 = new JLabel("Enter SGPA of V sem");
JLabel SGPA6 = new JLabel("Enter SGPA of VI sem");
JLabel SGPA7 = new JLabel("Enter SGPA of VII sem");
JLabel SGPA8 = new JLabel("Enter SGPA of VIII sem");
JLabel CGPA = new JLabel("CGPA obtained");

// Adding the textFeild
JTextField name = new JTextField(20);
JTextField usn = new JTextField(20);
JTextField age = new JTextField(3);
JTextArea address = new JTextArea(3, 4);
JComboBox<String> category = new JComboBox<String>();
JTextField cgpa = new JTextField(10);
JTextField sgpa1 = new JTextField(5);
JTextField sgpa2 = new JTextField(5);
JTextField sgpa3 = new JTextField(5);
JTextField sgpa4 = new JTextField(5);
JTextField sgpa5 = new JTextField(5);
JTextField sgpa6 = new JTextField(5);
JTextField sgpa7 = new JTextField(5);
JTextField sgpa8 = new JTextField(5);

JButton submit = new JButton("COMPUTE");
JButton done = new JButton("done");
JButton display = new JButton("display");
JTextArea stud_list_display = new JTextArea(20, 20);
JFrame f1 = new JFrame("Student Information");
JFrame f2 = new JFrame("Student Collection Display");

StudentClass() {// JFrame f1=new JFrame("Student Information");
  Name.setBounds(10, 10, 10, 10);
  category.addItem("GM");
  category.addItem("SC/ST");
  category.addItem("Cat1");
  category.addItem("Cat2");
  f1.add(Name);       f1.add(name);
  f1.add(USN);        f1.add(usn);
  f1.add(Age);        f1.add(age);
  f1.add(Address);    f1.add(address);
  f1.add(Category);   f1.add(category);
  f1.add(SGPA1);      f1.add(sgpa1);
  f1.add(SGPA2);      f1.add(sgpa2);
  f1.add(SGPA3);      f1.add(sgpa3);
  f1.add(SGPA4);      f1.add(sgpa4);
  f1.add(SGPA5);      f1.add(sgpa5);
  f1.add(SGPA6);      f1.add(sgpa6);
  f1.add(SGPA7);      f1.add(sgpa7);
  f1.add(SGPA8);      f1.add(sgpa8);
```

```
    f1.add(CGPA);       f1.add(cgpa);
    f1.add(submit);     f1.add(done);      f1.add(display);
    f2.add(stud_list_display);
    f1.setSize(900, 800);
    f1.setLayout(new GridLayout(8, 8));
    f1.setVisible(true);
    submit.addActionListener(this);
    done.addActionListener(this);
    display.addActionListener(this);
  }

  public void actionPerformed(ActionEvent event) {
    if (event.getSource() == submit) {
      try {
        int v1 = Integer.parseInt(age.getText());
        if (v1 < 18 || v1 > 35) {
          String age1 = JOptionPane.showInputDialog(null, "Enter valid Age");
          age.setText(age1);
        }
      } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(f1, "Invalid entry");
        age.requestFocus();
      }
      checkSGPA_valid(1, sgpa1);
      checkSGPA_valid(2, sgpa2);
      checkSGPA_valid(3, sgpa3);
      checkSGPA_valid(4, sgpa4);
      checkSGPA_valid(5, sgpa5);
      checkSGPA_valid(6, sgpa6);
      checkSGPA_valid(7, sgpa7);
      checkSGPA_valid(8, sgpa8);
      float calculate_CGPA = calculate_cgpa();
      cgpa.setText(Float.toString(calculate_CGPA));
    } else if (event.getSource() == done)// to submit into collection
    {
      Student newStudentObject = new Student(name.getText(), usn.getText(),
address.getText(),
          String.valueOf(category.getSelectedItem()), Integer.parseInt(age.getText()),
          Float.parseFloat(cgpa.getText()));
      studentList.add(newStudentObject);
    } else {
      f1.setVisible(false);
      f2.setVisible(true);
      f2.setSize(500, 500);
      stud_list_display.setText(" ");
      for (Student itr : studentList)
        stud_list_display.append(itr.toString() + "\n");
    }
  }

  float calculate_cgpa() {
    float v1 = Float.parseFloat(sgpa1.getText());
    float v2 = Float.parseFloat(sgpa2.getText());
    float v3 = Float.parseFloat(sgpa3.getText());
```

```
    float v4 = Float.parseFloat(sgpa4.getText());
    float v5 = Float.parseFloat(sgpa5.getText());
    float v6 = Float.parseFloat(sgpa6.getText());
    float v7 = Float.parseFloat(sgpa7.getText());
    float v8 = Float.parseFloat(sgpa8.getText());
    return (v1 + v2 + v3 + v4 + v5 + v6 + v7 + v8) / 8;
  }

  void checkSGPA_valid(int sem, JTextField sgpa) {
    try {
      if (Float.parseFloat(sgpa.getText()) > 10) {
        String v1 = JOptionPane.showInputDialog(null, "Enter an SGPA less than or equal to
10 for sem " + sem);
        sgpa.setText(v1);
      }
    } catch (NumberFormatException e) {
      String v2 = JOptionPane.showInputDialog(null, "Please enter SGPA for semester " +
sem);
      sgpa.setText(v2);
      // sgpa.requestFocus();
    }
  }

  public static void main(String args[]) {
    StudentClass sc = new StudentClass();
  }
}
```

Output :-