Abstract Classes

a. Bank accounts implementation

Aim:

To write a java program to use abstract class and implement functionality of a bank.

Code:

```
abstract class Accounts {
    double balance;
    int id;
    String name;
    String address;

    abstract double withdraw(double amount);
    abstract double deposit(double amount);
    void display() {
        System.out.println("ID: " + id);
        System.out.println("Name: " + name);
        System.out.println("Address: " + address);
        System.out.println("Balance: " + balance);
    }
}

class SavingsAccount extends Accounts {
    double withdraw(double amount){
        if(this.balance < amount){
            System.out.println("Insufficient Balance");
            return balance;
        }
        else{
            this.balance -= amount;
            return balance;
        }
    }
    double deposit(double amount) {
        balance += amount;
        return 0;
```

```
    }

    float rateOfInterest;
    SavingsAccount(int id, String name, String address, double
balance, float rateOfInterest){
        this.id = id;
        this.name = name;
        this.address = address;
        this.balance = balance;
        this.rateOfInterest = rateOfInterest;
    }

    double calculateInterest(int time){
        return balance * rateOfInterest * time;
    }
}

public class BankAccounts {
    public static void main(String[] args) {
        SavingsAccount savings = new SavingsAccount(123, "Joseph
Joestar", "123 Main St", 0, 0.0f);
        savings.deposit(1000);
        savings.display();
        savings.withdraw(500);
        savings.display();
    }
}
```

Output:

```
ID: 123
Name: Joseph Joestar
Address: 123 Main St
Balance: 1000.0
ID: 123
Name: Joseph Joestar
Address: 123 Main St
Balance: 500.0
```

b. Library implementation

Aim:

To write a java program to use abstract class and implement functionality of a library.

Code:

```
import java.util.HashMap;

abstract class LibraryManagement {
    HashMap<String, Integer> books = new HashMap<String, Integer>();
    void addBook(String bookName, int quantity){
        books.put(bookName, quantity);
    }
    void printBooks(){
        System.out.println("---Books---");
        for(String bookName: books.keySet()){
            System.out.println(bookName + ": " +
books.get(bookName));
        }
        System.out.println("-----------");
    }
    abstract void increment(String bookName);
    abstract void decrement(String bookName);
}

class Return_Issue extends LibraryManagement {
    void increment(String bookName){
        books.put(bookName, books.get(bookName) + 1);
        System.out.println("Book returned: " + bookName + "
Quantity: " + books.get(bookName));

    }
    void decrement(String bookName){
        books.put(bookName, books.get(bookName) - 1);
        System.out.println("Book issued: " + bookName + " Quantity:
" + books.get(bookName));
    }
    void search(String bookName){
        if(books.containsKey(bookName)){
            System.out.println("Book found");
            System.out.println("Book name: " + bookName);
```

```
            System.out.println("Quantity: " + books.get(bookName));
        }
        else{
            System.out.println("Book not found");
        }
    }
}

public class Library {
    public static void main(String[] args) {
        Return_Issue library = new Return_Issue();
        library.addBook("The Lord of the Rings", 5);
        library.addBook("The Hobbit", 3);
        library.addBook("The Silmarillion", 2);
        library.printBooks();
        library.search("The Hobbit");
        library.search("The Silmarillion");
        library.search("The Lord of the Rings");
        library.increment("The Hobbit");
        library.printBooks();
        library.decrement("The Silmarillion");
        library.printBooks();
    }
}
```

Output:

```
---Books---
The Silmarillion: 2
The Lord of the Rings: 5
The Hobbit: 3
-----------
Book found
Book name: The Hobbit
Quantity: 3
Book found
Book name: The Silmarillion
Quantity: 2
Book found
Book name: The Lord of the Rings
Quantity: 5
```

```
Book returned: The Hobbit Quantity: 4
---Books---
The Silmarillion: 2
The Lord of the Rings: 5
The Hobbit: 4
-----------
Book issued: The Silmarillion Quantity: 1
---Books---
The Silmarillion: 1
The Lord of the Rings: 5
The Hobbit: 4
-----------
```

Result:-

Thus, the programs were successfully executed and verified.