

# Abstract Class

---

1. Create an Abstract class Accounts with the following details

Data members:

1. Balance
2. Account number
3. Account holders name
4. Address

Methods:

1. withdrawal() - abstract
2. deposit() - abstract
3. display() - to show the balance of the account number

Create a subclass of this class SavingsAccount and add the following details

Data members:

1. rateofInterest

Methods:

2. calculateAmount()

**Aim:**

To implement a banking system.

**Code:**

```
package Exp8;

import java.util.Scanner;

public class Q1{
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter name: ");
        String name = input.nextLine();
        System.out.println("Enter account number: ");
        int accno = input.nextInt();
        System.out.println("Enter balance: ");
        float bal = input.nextFloat();
        System.out.println("Enter address: ");
        String add = input.nextLine();
        System.out.println("Enter rate of interest: ");
        float roi = input.nextFloat();
    }
}
```

```

        SavingsAccount myAcc = new SavingsAccount(name, accno, bal, add, roi);
        System.out.println("Enter 1 for withdrawal, 2 for deposit, 3 for
display, 4 for calculate amount");
        int choice = input.nextInt();
        switch(choice){
            case 1:
                System.out.println("Enter amount to withdraw");
                int amount = input.nextInt();
                myAcc.withdrawal(amount);
                break;
            case 2:
                System.out.println("Enter amount to deposit");
                amount = input.nextInt();
                myAcc.deposit(amount);
                break;
            case 3:
                myAcc.display();
                break;
            case 4:
                System.out.println("Enter time");
                int time = input.nextInt();
                System.out.println("Amount after " + time + " years: " +
myAcc.calculateAmount(time));
                break;
            default:
                System.out.println("Invalid choice");
        }
        input.close();
    }
}

abstract class Accounts {
    float balance;
    int accountNumber;
    String holderName, address;

    abstract void withdrawal(int amount);
    abstract void deposit(int amount);
    void display(){
        System.out.println("Account balance: " + balance);
    }
}

class SavingsAccount extends Accounts {
    float rateOfInterest;

    SavingsAccount(String name, int accno, float bal, String add, float roi) {
        super.holderName = name;
        super.balance = bal;
        super.accountNumber = accno;
        super.address = add;
        this.rateOfInterest = roi;
    }
}

```

```

float calculateAmount(int time) {
    return super.balance * (1 + this.rateOfInterest * time);
}

void withdrawal(int amount) {
    if (amount > super.balance) {
        System.out.println("Insufficient balance");
        return;
    }
    else {
        super.balance -= amount;
        System.out.println("Withdrawal successful");
        return;
    }
}

void deposit(int amount) {
    super.balance += amount;
    System.out.println("Deposit successful");
    return;
}
}

```

### Output:

```

Enter name:
Dhakkshin
Enter account number:
1234
Enter balance:
100000
Enter address:
qwe1
Enter rate of interest:
10
Enter 1 for withdrawal, 2 for deposit, 3 for display, 4 for calculate amount
4
Enter time
50
Amount after 50 years: 5.01E7

```

2. Create an Abstract class named Library Management that has functions for getting the book name and display the names of the books. There are two derived classes, Issue and Return. These classes have decrement and increment functions. The user can take or return books. Update the system accordingly. Include provision for searching for a book.

**Aim:**

To implement a library management system.

**Code:**

```
package Exp8;

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class Q2 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter 1 for issue, 2 for return, 3 for search");
        int choice = input.nextInt();
        input.nextLine(); // Consume newline

        switch (choice) {
            case 1:
                Issue issue = new Issue();
                issue.getBookName();
                issue.displayBookNames();
                break;
            case 2:
                Return ret = new Return();
                ret.getBookName();
                ret.displayBookNames();
                break;
            case 3:
                Search search = new Search();
                search.getBookName();
                break;
            default:
                System.out.println("Invalid choice");
        }

        input.close();
    }
}

abstract class LibraryManagement {
    protected List<String> books;

    public LibraryManagement() {
        books = new ArrayList<>();
        books.add("Book 1");
        books.add("Book 2");
        books.add("Book 3");
    }
}
```

```

    }

    abstract void getBookName();

    void displayBookNames() {
        System.out.println("Available books:");
        for (String book : books) {
            System.out.println(book);
        }
    }
}

class Issue extends LibraryManagement {
    Scanner input = new Scanner(System.in);

    void getBookName() {
        System.out.println("Enter book name to issue: ");
        String bookName = input.nextLine();

        if (books.contains(bookName)) {
            books.remove(bookName);
            System.out.println("You have borrowed " + bookName);
        }
        else {
            System.out.println(bookName + " is not available.");
        }
    }
}

class Return extends LibraryManagement {
    Scanner input = new Scanner(System.in);

    void getBookName() {
        System.out.println("Enter book name to return: ");
        String bookName = input.nextLine();

        books.add(bookName);
        System.out.println("You have returned " + bookName);
    }
}

class Search extends LibraryManagement {
    Scanner input = new Scanner(System.in);

    void getBookName() {
        System.out.println("Enter book name to search: ");
        String bookName = input.nextLine();

        if (books.contains(bookName)) {
            System.out.println(bookName + " is available.");
        }
        else {
            System.out.println(bookName + " is not available.");
        }
    }
}

```

```
}  
}
```

### Output:

```
Enter 1 for issue, 2 for return, 3 for search  
3  
Enter book name to search:  
Book 3  
Book 3 is available.  
  
Enter 1 for issue, 2 for return, 3 for search  
1  
Enter book name to issue:  
Book 3  
You have borrowed Book 3  
Available books:  
Book 1  
Book 2  
  
Enter 1 for issue, 2 for return, 3 for search  
2  
Enter book name to return:  
Book 3  
You have returned Book 3  
Available books:  
Book 1  
Book 2  
Book 3  
Book 3
```

---

### Result:

All the programs are executed and the output are verified.

---

---