# Assignment. 10

**Title:** Case **Study**

**Aim:** Using conceptsof Object Oriented programming develop solutionfor any one application

1. Banking solution contains followingoperations such as 1. Create an account 2.Deposit money 3.Withdraw money 4. Honor daily withdrawal limit 5. Check the balance 6. Display Account information.
2. Inventory management contains followingoperations such as List of all products 2.

Display individual

product information 3. Purchase 4. Shipping 5.Balance stock 6. Loss and Profit calculation.

**Objectives: To implement real time context. Sample Code:**

Create following classes and Methods

class Account : Set balance(),Getbalance(),setAccount type(),getaccounttype Class Bank: creatAccount(),withdrawAmount(),depositAmount(),displayinfo()

Class customer: getCustomerName(),setCustomerName(),getcustomerAge(),setCustomerAge() Class saving account: setMinimumBalance (),withdraw()

**Input:**

Enter your name: Sai Enter your age: 15

Minimum age should be 18 to create an account. Please enter valid age: 21

Enter your account Id: 1

Enter your account type: savings Enter balance: 10000

Enter minimum balance: 1000

**Output :**

1. Create Account 2.Display Account 3.Check Balance 4.Deposit Amount 5.Withdraw Amount Enter your choice: 1 Enter your name: Sai Enter your age: 15

Minimum age should be 18 to create an account. Please enter valid age: 21

Enter your account Id: 1

Enter your account type: savings Enter balance: 10000

Enter minimum balance: 1000

Do you want to perform more actions? (yes/no): yes 1.Create Account

1. Display Account 3.Check Balance 4.Deposit Amount 5.Withdraw Amount Enter your choice: 2

Welcome Sai Pande! Following are your account details: Age :21

Account Id: 1

Account Type: savings Balance: 10000.0

Minimum balance: 1000.0

Do you want to perform more actions? (yes/no): yes 1.Create Account

2.Display Account 3.Check Balance 4.Deposit Amount 5.Withdraw Amount Enter your choice: 3 Balance is: 10000.0

Do you want to perform more actions? (yes/no): yes 1.Create Account

2.Display Account 3.Check Balance 4.Deposit Amount 5.Withdraw Amount Enter your choice: 4

Enter the amount you want to deposit: 20000 Amount deposited successfully. Balance is: 30000.0 Do you want to perform more actions? (yes/no): yes 1.Create Account

2.Display Account 3.Check Balance 4.Deposit Amount 5.Withdraw Amount Enter your choice: 5

Enter the amount you want to withdraw: 30000

Withdrawal failed. Maximum limit of withdrawal in one transaction is Rs.20000. Do you want to perform more actions? (yes/no): yes

1. Create Account 2.Display Account 3.Check Balance 4.Deposit Amount 5.Withdraw Amount Enter your choice: 5

Enter the amount you want to withdraw: 15000 Withdrawal successful. Balance is: 15000.0

ALGORITHM

1 By using constructors we have declared many user defined functions and ased the user to enter the bank details

2 Created a saving account class and asked the user to enter the following details.

3 Similarly created a class for withdrawal and deposit of the money in the bank.

4 In the Main class declaring all the bank functions in a switch case type of menu.

5 End.

CODE:

package com.company;  
  
import java.util.Scanner;  
  
class Customer{  
 private String customerName;  
 private int customerAge;  
  
 public void setCustomerName(String customerName) {  
 this.customerName = customerName;  
 }  
  
 public String getCustomerName(){  
 return customerName;  
 }  
  
 public void setCustomerAge(int customerAge){  
 this.customerAge=customerAge;  
 }  
 public int getCustomerAge(){  
 return customerAge;  
 }  
}  
  
abstract class Account{  
 protected double balance ;  
 protected int accountId;  
 protected String accountType; *// here is account type* protected Customer custObj;  
  
 void setBalance(double balance){  
 this.balance = balance;  
 }  
 double getBalance(){  
 return balance;  
 }  
  
 void setAccountId(int accountId) {  
 this.accountId = accountId;  
 }  
 int getAccountId(){  
 return accountId;  
 }  
  
 void setAccountType(String accountType) {  
 this.accountType = accountType;  
 }  
  
 String getAccountType(){*//accounttype* return accountType;  
 }  
  
 void setCustObj(Customer custObj) {  
 this.custObj = custObj;  
 }  
 Customer getCustObj(){  
 return custObj;  
 }  
  
 public abstract boolean withdraw(double amount);  
}  
  
class SavingAccount extends Account{  
 private double minBalance;  
 public void setMinBalance(double minBalance){  
 this.minBalance = minBalance;  
 }  
 public double getMinBalance(){  
 return minBalance;  
 }  
  
 public boolean withdraw(double amount){  
 if((balance-amount)>minBalance){  
 balance-=amount;  
 return true;  
 }  
 else  
 return false;  
 }  
}  
  
  
class Bank{  
 Scanner in = new Scanner(System.*in*);  
  
 public SavingAccount a = new SavingAccount();  
 public Customer c =new Customer();  
  
 public SavingAccount createAccount(){  
 in.nextLine();  
  
 System.*out*.print("Enter your name = ");  
 String customername = in.nextLine();  
 c.setCustomerName(customername);  
  
 System.out.print("Enter your age = ");  
 int customerAge = in.nextInt();  
 if(customerAge<18){  
 do{  
 System.out.println("Minimum age should be 18 to create a bank account !");  
 customerAge = in.nextInt();  
 }while(customerAge<18);  
 }  
 c.setCustomerAge(customerAge);  
  
 a.setCustObj(c);  
 System.out.print("Enter your account ID = ");  
 int accountID = in.nextInt();  
 a.setAccountId(accountID);  
  
  
 System.out.print("Enter your account type = ");  
 String accountype = in.next();  
 a.setAccountType(accountype);  
  
  
 System.out.print("\nEnter the balance = ");  
 double balance = in.nextDouble();  
 a.setBalance(balance);  
  
 System.out.print("Enter the minimum balance = ");  
 double minbalance = in.nextDouble();  
 a.setMinBalance(minbalance);  
  
 return a;  
 }  
void getWithdrawAmount() {  
 System.out.print("Enter the amount you want to withdraw = ");  
 double amount = in.nextDouble();  
 if (amount > 20000) {  
 System.out.println("Withdrawal failed . Maximum limit of one withdrawal is Rs. 20000.");  
 } else {  
 if (a.withdraw(amount)) {  
 System.out.println("Withdrawal successfully done . " + a.getBalance());  
 } else {  
 System.out.println("Sorry !! Not enough balance .");  
 }  
 }  
}  
 public void depositAmount(double amount){  
 double bal = a.getBalance()+amount;  
 a.setBalance(bal);  
 System.out.println("Amount deposited successfully . Balance is = "+a.getBalance());  
 }  
 public void checkBalance(){  
 System.out.println("Balance is = "+a.getBalance());  
 }  
  
 public void displayAccountInfo(){  
 System.out.println("Welcome account holder :) "+c.getCustomerName()+" Following are your account details .");  
 System.out.println("Age = "+c.getCustomerAge());  
 System.out.println("Account ID = "+a.getAccountId());  
 System.out.println("Account type = "+a.getAccountType());  
 System.out.println("Account Balance = "+a.getBalance());  
 System.out.println("Account Minimum Balance = "+a.getMinBalance());  
 }  
  
}  
  
public class Assignmnetn09 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.in);  
 SavingAccount a ;  
 Bank bm = new Bank();  
  
 do{  
 System.out.println("\n\t1.Create Account\n\t2.Display Account\n\t3.Check Balance\n\t4.Deposit Amount\n\t5.Withdraw Amount\n\t6.Exit");  
 System.out.print("Enter your choice = ");  
 int choice = in.nextInt();  
 System.out.println("");  
  
 switch (choice) {  
 case 1 -> {  
 a = bm.createAccount();  
 System.out.println("------------------------------------------------");  
 }  
 case 2 -> {  
 bm.displayAccountInfo();  
 System.out.println("------------------------------------------------");  
 }  
 case 3 -> {  
 bm.checkBalance();  
 System.out.println("------------------------------------------------");  
 }  
 case 4 -> {  
 System.out.print("Enter the amount you deposit = ");  
 double amt = in.nextDouble();  
 bm.depositAmount(amt);  
 System.out.println("------------------------------------------------");  
 }  
 case 5 -> {  
 bm.getWithdrawAmount();  
 System.out.println("------------------------------------------------");  
 }  
 case 6 -> {  
 System.out.println("------------------------------------------------");  
 return;  
 }  
 default -> {  
 System.out.println("Invalid Input !");  
 System.out.println("-------------------------------------------------");  
 }  
 }  
 }while(true);  
  
  
 }  
}

OUTPUT:





CONCLUSION

IN THIS ASSIGNMENT WE HAVE SUCCESSFULLY LEARNED THE CONCEPT OF OOP IN JAVA.

Frequently Ask Question:

* 1. List the features which are used for application development
  2. How the polymorphism applied
  3. Can we apply interface or abstract class in given case study ? How.
  4. Did the application takes care of garbage collection
  5. Have you applied user defined exceptions in given case study ? Give examples .
  6. How many objects are created and how they are stored in memor