

Week-8

Page No.:	1
Date :	/ /

Program - 1

```

import java.util.Scanner;
class First {
    abstract class Shape {
        int b, h;
        Shape (int b, int h) {
            this. b = b;
            this. h = h;
        }
        void printArea () {
        }
    }
    class Rectangle extends Shape {
        Rectangle (int b, int h) {
            super (b, h);
        }
        void printArea () {
            System.out.println ("Area of the rectangle is " +
                (b*h));
        }
    }
    class Triangle extends Shape {
        Triangle (int b, int h) {
            super (b, h);
        }
        void printArea () {
            System.out.println ("Area of the triangle is " +
                ((b*h)/2));
        }
    }
}

```

```
Circle (int) r;
    Super (0, r);
}
```

```
void printArea() {
    System.out.println ("Area of the circle is " +  

        (Math.PI * h * w));
}
}
```

```
public class Program {
    public static void main (String [] args) {
        Triangle tri = new Triangle (10, 5);
        Rectangle rect = new Rectangle (10, 5);
        Circle c = new Circle (10);
        rect.printArea ();
        tri.printArea ();
        c.printArea ();
    }
}
```

if you want to print rectangle area
 $((d * d))$

I added another statement
 $System.out.println ("Area of the triangle is " +$
 $((d * d))$)

(1) 2020 January Lecture
 6.1 with both part (a) printing two, one
 $((d * d))$,

```
Prog1 - Notepad
File Edit Format View Help
import java.util.Scanner;

abstract class Shape{
    int b,h;
    Shape(int b, int h){
        this.b = b;
        this.h = h;
    }
    void printArea(){};
}

class Rectangle extends Shape{
    Rectangle(int b, int h){
        super(b,h);
    }

    void printArea() {
        System.out.println("Area of the rectangle is " + (b*h));
    }
}

class Triangle extends Shape{
    Triangle(int b, int h){
        super(b,h);
    }

    void printArea() {
        System.out.println("Area of the triangle is " + ((b*h)/2));
    }
}

class Circle extends Shape{
    Circle(int r){
        super(0,r);
    }

    void printArea() {
        System.out.println("Area of the circle is " + (Math.PI*r*r));
    }
}

public class Prog1{
    public static void main(String[] args) {
        Rectangle rect = new Rectangle(10, 5);
        Triangle tri = new Triangle(10, 5);
        Circle c = new Circle(10);
        rect.printArea();
        tri.printArea();
        c.printArea();
    }
}
```

```
Command Prompt
Microsoft Windows [Version 10.0.18363.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\91957>cd desktop
C:\Users\91957\Desktop>cd java lab
C:\Users\91957\Desktop\java lab>javac Prog1.java
C:\Users\91957\Desktop\java lab>java Prog1
Area of the rectangle is 50
Area of the triangle is 25
Area of the circle is 314.1592653589793
C:\Users\91957\Desktop\java lab>
```

Program-2

```
import java.util.Scanner;
```

abstract class Account {

```
String cName, accNo, accType;
public static final String ASS1 - RED = "1001B[31m";
public static final String ASS1 - GREEN = "1001B[32m";
public static final String ASS1 - RESET = "1001B[0m";
Scanner sc = new Scanner (System.in);
```

```
Account (String name, String accNo, String accType) {
    this.cName = name;
    this.accNo = accNo;
    this.accType = accType;
}
Account () {};
```

```
class CurrentAcc extends Account {
    private double balance = 5000, rate = 0.06;
```

```
CurrentAcc (String name, String accNo, String accType) {
    super (name, accNo, accType);
    System.out.println ("Welcome" + cName);
}
```

```
void getBalance () {
    System.out.format ("Your balance : %f\n", balance);
}
void deposit (double amount) {
    choice;
```

```
System.out.println ("Deposit. Account holder: " + cName  
+ "Amount: " + amount);  
System.out.println ("Approve Deposit? (Y/N):");  
choice = sc.nextLine().charAt(0);  
  
if (choice == 'Y') {  
    balance += amount;  
    System.out.println ("Deposit approved.  
Updated balance: " + balance + ANSS_RESET);  
}  
  
else {  
    System.out.println (ANSI_RED + "Deposit not approved"  
+ ANSI_RESET);  
}  
  
void withdrawl (double amount) {  
    if (amount > balance) {  
        System.out.println ("This account cannot  
withdraw any funds" + ANSI_RESET);  
    }  
  
    void calcInterest () {}  
    void checkMinAmount () {  
        if (balance < 3000) {  
            balance -= 500;  
        System.out.println (ANSI_RED + "Balance under  
minimum amount to be maintained." +  
ANSI_RESET);  
    }  
}  
}
```

```
class SavingsAcc extends Account {
    private double balance = 5000, rate = 0.06;

    SavingsAcc(String name, String accNo, String accType) {
        super(name, accNo, accType);
        System.out.println("New customer: " + cName);
    }

    void getBalance() {
        System.out.format("Your balance: %f \n", balance);
    }

    void deposit(double amount) {
        choice;
        System.out.println("Deposit. Account holder: " + cName +
                           " Amount: " + amount);
        System.out.println("Approved Deposit? (y/n): ");
        choice = sc.next();
        choice = choice.charAt(0);
        if (choice == 'y' || choice == 'Y') {
            balance += amount;
            System.out.println("ANSWER-GREEN + \"Deposit approved.\"
                               updated balance: " + balance + ANSWER_RESET);
            calcInterest();
            checkMinAmount();
        } else {
            System.out.println("ANSWER-RED + \"Deposit not approved\" +
                               ANSWER_RESET");
        }
    }

    void calculate() {
        double cur;
```

```

    CC = balance * (Math. pow((1 + rate), 10), 2);
    balance += CC;
    System.out.println("ANSI - GREEN + " + interest);
    updatedBalance := +balance + ANSC - RESET;
}

void withdraws(double amount) {
    if (amount > balance) {
        choice = 0;
        System.out.println("Insufficient funds");
        return;
    }
    System.out.println("Amount : " + amount);
    System.out.println("ANSI - RED + " + accountBalance);
    if (amount < withdraw + lower than amount to be withdrawn + ANSC - RESET);
    else {
        choice = 1;
        System.out.println("APPROVE" + cName + "'s request for withdrawal");
        if (choice == 'Y') {
            balance -= amount;
            System.out.println("ANSI - GREEN + Withdrawal approved.");
            updatedBalance := +balance + ANSC - RESET;
            calculateInterest();
            checkMinAmount();
        } else {
            System.out.println("ANSI - RED + Withdrawal not approved");
        }
    }
}

void checkMinAmount() {
    int minAmount = 3000, penalty = 500;
    if (balance < minAmount) {
        System.out.println("Penalty : " + penalty);
    }
}

```

balance = penalty;

```
System.out.println("ANSI - RECD + " + balance + " minimum amount  
to be maintained." + ANSI - RESET);  
System.out.println("ANSI - FCD + " + penalty imposed. Updated  
balance; " + balance + ANSI - RESET);  
}  
}  
public class Main {  
    public static void main (String [] args) {  
        int i;  
        double temp;  
  
        String name, accNo, accType;  
        Scanner sc = new Scanner (System.in);  
  
        System.out.println ("Enter Name:");  
        name = sc.nextLine();  
        System.out.println ("Enter Account number:");  
        accNo = sc.nextLine();  
        System.out.println ("Enter Account Type:");  
        accType = sc.nextLine();  
  
        if (accType.charAt(0) == 'C') {  
            CurrentAcc a = new CurrentAcc (name, accNo, accType);  
            while (true) {  
                System.out.println ("1. Deposit money\n 2. withdraw money  
 3. Display money info.\n Enter choice: ");  
                c = sc.nextInt();  
                switch (c) {
```

Case 1: d

```

from os import system
temp = sc.nextDouble();
temp = sc.nextDouble();
a. deposit (temp);
break;
}

```

Case 2: d

```

System.out.println ("Enter amount to be withdrawn");
temp = sc.nextDouble();
a.withdraw (temp);
break;
}

```

Case 3: d

```

a.getBalance();
break;
}

```

Case 4: d

```

System.exit(0);
break;
}

```

Default : System.out.println ("Enter the correct option");

```

else if (accType.charAt(0) == 'S') {
    SavingsAcc a = new SavingsAcc (name, accNo, accType);
    while (true) {
        System.out.println ("1. Deposit money\n2. withdraw money
                            \n3. Display money\n4. Exit");
        c = sc.nextInt();
        switch (c) {
            case 1:
                a.deposit ();
                break;
            case 2:
                a.withdraw ();
                break;
            case 3:
                a.display ();
                break;
            case 4:
                System.exit(0);
                break;
            default:
                System.out.println ("Enter the correct option");
        }
    }
}
```

```
System.out.println("Enter amount to be deposited: ");
temp = Sc.nextInt();
a. deposit(temp);
break;
}

Case 2: if
System.out.println("Enter amount to be withdraw:");
temp = Sc.nextInt();
a. withdraw(temp);
break;
}

Case 3: if
a. getBalance();
break;
}

Case 4: if
System.out.println("Enter the contact options");
temp = Sc.nextInt();
a. defaultOption();
break;
}

else {
System.out.println("Enter valid type ... exiting");
}
```

```
*Prog1 - Notepad
File Edit Format View Help
import java.util.Scanner;

abstract class Account{
    String cName,accNum,accType;
    public static final String ANSI_RED = "\u001B[31m";
    public static final String ANSI_GREEN = "\u001B[32m";
    public static final String ANSI_RESET = "\u001B[0m";
    Scanner sc = new Scanner(System.in);

    Account(String name,String accNo,String accType){
        this.cName = name;
        this.accNum = accNo;
        this.accType = accType;
    }
    Account(){}
}

class CurrentAcc extends Account{
    private double balance = 5000,rate = 0.05;
    CurrentAcc(String name,String accNo,String accType){
        super(name,accNo,accType);
        System.out.println("Welcome " + cName);
    }
    void getBalance() {
        System.out.format("Your balance: %f\n",balance);
    }
    void deposit(double amount){
        char choice;
        System.out.println("Deposit. Account holder: " + cName + " Amount: " + amount);
        System.out.println("Approve Deposit?(Y/N): ");
        choice = sc.next().charAt(0);
        if(choice == 'Y' || choice == 'y'){
            balance+=amount;
            System.out.println(ANSI_GREEN + "Deposit approved. Updated balance: " + balance + ANSI_RESET);
        }else{
            System.out.println(ANSI_RED + "Deposit not approved" + ANSI_RESET);
        }
    }
    void withdraw(double amount){
        System.out.println(ANSI_RED + "This account cannot withdraw any funds" + ANSI_RESET);
    }
    void calcInterest() {};
    void checkMinAmount(){
        if(balance < 3000){
            balance+=500;
            System.out.println(ANSI_RED + "Balance under minimum amount to be maintained." + ANSI_RESET);
            System.out.println(ANSI_RED + "Penalty imposed. Updated balance: " + balance + ANSI_RESET);
        }
    }
}

Type here to search 0 16:16 03-11-2020
```

```
*Prog1 Notepad
File Edit Format View Help
]

class SavingsAcc extends Account{
    private double balance = 5000,rate = 0.05;
    SavingsAcc(String name, String accNo, String accType){
        super(name, accNo, accType);
        System.out.println("New Customer: " + cName);
    }
    void getBalance(){
        System.out.format("Your balance: %f\n",balance);
    }
    void deposit(double amount){
        char choice;
        System.out.println("Deposit. Account holder: " + cName + " Amount: " + amount);
        System.out.println("Approve Deposit? (Y/N): ");
        choice = sc.next().charAt(0);
        if(choice == 'Y' || choice == 'y'){
            balance+=amount;
            System.out.println(ANSI_GREEN + "Deposit approved. Updated balance: " + balance + ANSI_RESET);
            calcInterest();
            checkMinAmount();
        }else{
            System.out.println(ANSI_RED + "Deposit not approved" + ANSI_RESET);
        }
    }
    void calcInterest(){
        double CI;
        CI = balance * (Math.pow((1+rate/100),2));
        balance+=CI;
        System.out.println(ANSI_GREEN + "Interest added. Updated balance: " + balance + ANSI_RESET);
    }
    void withdraw(double amount){
        char choice;
        if(balance < amount){
            System.out.println(ANSI_RED + "Account balance is lower than amount to be withdrawn" + ANSI_RESET);
            return;
        }
        System.out.println("Approve " + cName + "'s request for withdrawal? (Y/N): ");
        choice = sc.next().charAt(0);
        if(choice == 'Y' || choice == 'y'){
            balance-=amount;
            System.out.println(ANSI_GREEN + "Withdrawal approved. Updated balance: " + balance + ANSI_RESET);
            calcInterest();
            checkMinAmount();
        }else{
            System.out.println(ANSI_RED + "Withdrawal not approved" + ANSI_RESET);
        }
    }
}
```

Ln 210, Col 1 | 80% | Windows (CRLF) | UTF-8

Q E M F X ENG 03-11-2020

Prog2 Notepad

```

File Edit Format View Help

if(account.charAt(0) == 'c'){
    CurrentAcc c = new CurrentAcc(name, accno, accType);
    while(true){
        System.out.println("1. Deposit money\n2. Withdraw money\n3. Display money\n4. Exit");
        c = select(c);
        switch(c){
            case 1:
                System.out.print("Enter amount to be deposited: ");
                temp = sc.nextInt();
                a.deposit(temp);
                break;
            case 2:
                System.out.print("Enter amount to be withdrawn: ");
                temp = sc.nextInt();
                a.withdraw(temp);
                break;
            case 3:
                a.getBalance();
                break;
            case 4:
                System.exit(0);
                break;
            default: system.out.println("Enter the correct options");
        }
    }
}

else if(account.charAt(0) == 's'){
    SavingsAcc s = new SavingsAcc(name, accno, accType);
    while(true){
        System.out.println("1. Deposit money\n2. Withdraw money\n3. Display money\n4. Exit");
        c = sc.nextInt();
        switch(c){
            case 1:
                System.out.print("Enter amount to be deposited: ");
                temp = sc.nextInt();
                a.deposit(temp);
                break;
            case 2:
                System.out.print("Enter amount to be withdrawn: ");
                temp = sc.nextInt();
                a.withdraw(temp);
                break;
            case 3:
                a.getBalance();
                break;
            case 4:
                System.exit(0);
                break;
            default: System.out.println("Enter the correct options");
        }
    }
}

else{
    System.out.println("Enter valid type... exiting");
}
}

```

Command Prompt

```

Microsoft Windows [Version 10.0.18363.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\91957>cd desktop

C:\Users\91957\Desktop>cd java lab

C:\Users\91957\Desktop\java lab>javac Prog2.java

C:\Users\91957\Desktop\java lab>java Prog2
Enter Name:
abc
Enter Account number:
18493bg
Enter Account Type:
ghi
Enter valid type... Exiting

C:\Users\91957\Desktop\java lab>_

```

Ln 182, Col 13 | 60% | Windows (CRLF) | UTF-8

1621 03-11-2020