1) - Create a class DateTime containing private fields: Day, Month, Year, Hour, Minutes, Seconds.

**Constructors:**  
(i) DateTime()  
(ii) DateTime(dd,mm,yyyy)  
(iii) DateTime(dd,mm,yyyy,hh,mm,ss)

**Methods:**

(i) **String toDateString()** - will return the date as string in the format - dd/mm/yyyy hh:mm:ss

(ii) **String toDateString(String Format)**  
- if Format is "DM" then return the date string in the format - mm/dd/yyyy  
- if "DD" - dd/mm/yyyy  
- if "TS" - hh:mm  
- if "TL" - hh:mm:ss  
- "DMTS" - mm/dd/yyyy hh:mm  
- Other valid formats are : "DMTL", "DDTS", "DDTL". Rest are invalid.

Program –

import java.util.Scanner;

class P2{

public static void main(String args[])

{

Scanner sc = new Scanner( System.in );

int x,y,z,a,b,c;

System.out.println("Enter date (dd mm yyyy hh mm ss)");

x = sc.nextInt();

y = sc.nextInt();

z = sc.nextInt();

a = sc.nextInt();

b = sc.nextInt();

c = sc.nextInt();

DateTime d = new DateTime(x,y,z,a,b,c);

System.out.println(d.toDateString());

System.out.println(d.toDateString(“DM”));

}

}

class DateTime{

private int day,month,yr,hr,min,sec;

DateTime(){

day = month = yr = hr = min = sec = 0;

}

DateTime(int x,int y,int z){

day = x;

month = y;

yr = z;

hr = min = sec = 0;

}

DateTime(int x,int y,int z,int a,int b,int c){

day = x;

month = y;

yr = z;

hr = a;

min = b;

sec = c;

}

public String toDateString()

{

String s = day + "/" + month + "/" + yr + " " + hr + ":" + min + ":" + sec;

return s;

}

public String toDateString(String Format)

{

String s;

if(Format == "DM")

s = month + "/" + day + "/" + yr;

else if(Format == "DD")

s = day + "/" + month + "/" + yr;

else if(Format == "TS")

s = hr + ":" + min;

else if(Format == "TL")

s = hr + ":" + min + ":" + sec;

else if(Format == "DMTS")

s = month + "/" + day + "/" + yr+ " " + hr + ":" + min;

else if(Format == "DMTL")

s = month + "/" + day + "/" + yr+ " " + hr + ":" + min + ":" + sec;

else if(Format == "DDTS")

s = day + "/" + month + "/" + yr + " " + hr + ":" + min;

else if(Format == "DDTL")

s = day + "/" + month + "/" + yr + " " + hr + ":" + min + ":" + sec;

}

}

Output -

Enter date (dd mm yyyy hh mm ss)

10 04 1996 04 54 12

10/4/1996 4:54:12

4/10/1996

2)Create a class Person with the following properties:

**Private fields:** ID, Name, YearOfBirth

Now take input (ID, Name, YearOfBirth) from the user and print it in the format :

ID | Name | Age

e.g. for ID:

**Declaration in the class :** private int ID;  
**get method:** int getID(){return ID;}  
**set method:** int setID(int id) {ID = id);

Do this for all the fields.

Define one more method **getAge()** in Person class which will return the age of the person.

Define the following constructors in the Person class

**Person(int ID, String Name, int YearOfBirth)  
Person(String Name, int YearOfBirth)**- generate the ID randomly in this case.

Create the following method in the Person class:

**void CopyTo(Person p)** - It will copy the fields of current(or this) Person into the person p.

Program –

import java.util.Scanner;

import java.util.Calendar;

import java.lang.\*;

class Person

{

private int Id;

private String Name;

private int YearOfBirth;

int getId(){

return Id;

}

String getName(){

return Name;

}

int getYear(){

return YearOfBirth;

}

void setId(int x){

Id = x;

}

void setName(String x){

Name = x;

}

void setYear(int x){

YearOfBirth = x;

}

int getAge(){

int year = Calendar.getInstance().get(Calendar.YEAR);

return (year - YearOfBirth);

}

Person(int Id, String Name, int YearOfBirth){

setId(Id);

setName(Name);

setYear(YearOfBirth);

}

Person(String Name, int YearOfBirth){

int x = (int) Math.floor(Math.random() \* 101);

setId(x);

setName(Name);

setYear(YearOfBirth);

}

void CopyTo(Person p){

p.setId(Id);

p.setName(Name);

p.setYear(YearOfBirth);

}

}

class P4{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int a,c;

String b;

Person p = new Person(1,”a”,1996);

System.out.println("Enter ID:");

p.setId(sc.nextInt());

System.out.println("Enter Name:");

p.setName(sc.next());

System.out.println("Enter YearOfBirth:");

p.setYear(sc.nextInt());

System.out.println(p.getId() + " | " + p.getName() + " | " + p.getAge());

System.out.println("Next person");

System.out.println("Enter ID:");

a = sc.nextInt();

System.out.println("Enter Name:");

b = sc.next();

System.out.println("Enter YearOfBirth:");

c = sc.nextInt();

p = new Person(a,b,c);

System.out.println("Next person");

System.out.println("Enter Name:");

b = sc.next();

System.out.println("Enter YearOfBirth:");

c = sc.nextInt();

Person q = new Person(b,c);

System.out.println(p.getId() + " | " + p.getName() + " | " + p.getAge());

System.out.println(q.getId() + " | " + q.getName() + " | " + q.getAge());

}

}

Output -

Enter ID:

12

Enter Name:

Hillary

Enter YearOfBirth:

1947

12 | Hillary | 69

Next person

Enter ID:

13

Enter Name:

Bernie

Enter YearOfBirth:

1941

Next person

Enter Name:

Donald

Enter YearOfBirth:

1946

13 | Bernie | 75

79 | Donald | 70

3) Create a **BankAccount**class.

Some of the fields can be **CustomerID, Amount, AccountType (Current, Saving).**

The customer must deposit some amount (>= Rs 1000) while opening the account. The account can never have Zero balance.   
The valid operations are **deposit, withdraw, checkBalance, closeAccount.**

Add field**password**.

Add the following restrictions:

(i) A current/saving account holder should not be able to deposit more than Rs 5000/Rs 10000 at a time respectively.

(ii) if three successive withdraw operations are performed at a time. The account should get Blocked. For any next Withdraw user should be asked to enter the password. After that account should again become unblocked.

Program –

import java.util.Scanner;

class BankAccount{

private int CustomerId;

private float Amount;

private char AccountType;

private String password;

private int withdraw;

BankAccount( int id, float amount, char type, String passcode){

Amount = amount;

AccountType = type;

CustomerId = id;

password = passcode;

withdraw = 0;

}

void withdraw(float x){

Scanner st = new Scanner(System.in);

String pass = password;

if(withdraw >= 3)

{

System.out.println("Enter password");

pass = st.next();

if(pass.equals(password))

withdraw = 0;

}

if(Amount - x > 1000 && pass.equals(password))

{

Amount -= x;

System.out.println("Transaction Successful");

withdraw ++;

}

else

{

System.out.println("Transaction Failed");

withdraw ++;

}

}

void deposit(float x){

withdraw = 0;

float lim = 0;

if(AccountType == 'c')

lim = 5000;

else

lim = 10000;

if(x <= lim){

Amount += x;

System.out.println("Transaction Successful");

}

else

System.out.println("Transaction Failed");

}

float checkBalance(){

withdraw = 0;

return Amount;

}

void closeAccount(){

withdraw = 0;

Amount = 0;

}

void disp(){

withdraw = 0;

System.out.println("ID: " + CustomerId + "\n" + "Amount: " + Amount);

}

}

class P5{

public static void main(String args[])

{

int x = 0;

int ab=0;

float b=0;

char c=0;

String d = "";

mport java.util.Scanner;

class BankAccount{

private int CustomerId;

private float Amount;

private char AccountType;

private String password;

private int withdraw;

BankAccount( int id, float amount, char type, String passcode){

Amount = amount;

AccountType = type;

CustomerId = id;

password = passcode;

withdraw = 0;

}

void withdraw(float x){

Scanner st = new Scanner(System.in);

String pass = password;

if(withdraw >= 3)

{

System.out.println("Enter password");

pass = st.next();

if(pass.equals(password))

withdraw = 0;

}

if(Amount - x > 1000 && pass.equals(password))

{

Amount -= x;

System.out.println("Transaction Successful");

withdraw ++;

}

else

{

System.out.println("Transaction Failed");

withdraw ++;

}

}

void deposit(float x){

Scanner sc = new Scanner( System.in );

BankAccount B;

while( x == 0 )

{

System.out.println("Enter account no:");

ab = sc.nextInt();

System.out.println("Enter account type:");

c = sc.next().charAt(0);

System.out.println("Enter amount:");

b = sc.nextFloat();

System.out.println("Enter password:");

d = sc.next();

if(b > 1000)

x = 1;

}

B = new BankAccount(ab,b,c,d);

while(x == 1)

{

System.out.println("1) Deposit\n2) Withdraw\n3) Check Balance\n4) Display Details\n5) Close Account\n6) Exit");

int a = sc.nextInt();

switch(a)

{

case 1: System.out.println("Enter amount");

B.deposit(sc.nextFloat());

break;

case 2: System.out.println("Enter amount");

B.withdraw(sc.nextFloat());

break;

case 3: System.out.println("The balance is " + B.checkBalance());

break;

case 4: B.disp();

break;

case 5: B.closeAccount();

break;

case 6: System.exit(0);

}

}

}

}

Output -

Enter account no:

12

Enter account type:

c

Enter amount:

1200

Enter password:

friedchicken

1) Deposit

2) Withdraw

3) Check Balance

4) Display Details

5) Close Account

6) Exit

1

Enter amount

40

Transaction Successful

1) Deposit

2) Withdraw

3) Check Balance

4) Display Details

5) Close Account

6) Exit

2

Enter amount

30

Transaction Successful

1) Deposit

2) Withdraw

3) Check Balance

4) Display Details

5) Close Account

6) Exit

2

Enter amount

30

Transaction Successful

1) Deposit

2) Withdraw

3) Check Balance

4) Display Details

5) Close Account

6) Exit

2

Enter amount

30

Transaction Successful

1) Deposit

2) Withdraw

3) Check Balance

4) Display Details

5) Close Account

6) Exit

2

Enter amount

30

Enter password

friedchicken

Transaction Successful

1) Deposit

2) Withdraw

3) Check Balance

4) Display Details

5) Close Account

6) Exit

3

The balance is 1120.0

1) Deposit

2) Withdraw

3) Check Balance

4) Display Details

5) Close Account

6) Exit

4

ID: 12

Amount: 1120.0

1) Deposit

2) Withdraw

3) Check Balance

4) Display Details

5) Close Account

6) Exit

5

1) Deposit

2) Withdraw

3) Check Balance

4) Display Details

5) Close Account

6) Exit

6

4) Create a base class Shape with the following methods:

**double getArea() -** return the area of the shape.  
**double getPerimeter() -** return the perimeter of the shape.  
**void Draw() -** print the shape type like circle, rectangle etc.

The definitions of these functions in**Shape** class can include some default logic. e.g. in this case you can return a 0 from the first two methods and print some default value in 3rd method.

Now derive two classes from the Shape class.  
**Circle** with a field Radius  
**Rectangle** with fields length and breadth.  
These classes should override the methods defined in the Shape class.

Create a menu driven program to allow user to create new shapes and do operations on them.

Now derive a class**Square** from the class**Rectangle.**  
From the constructor of Square class, call the constructor of Rectangle class.  
Add the option of creating a **Square** in your menu.

Derive a class Cylinder from Circle.  
Add the option of creating a **Cylinder** in your menu.

Program –

import java.util.Scanner;

class Shape{

double getArea()

{

return 0;

}

double getPerimeter()

{

return 0;

}

void draw()

{

System.out.println("Shape not defined");

}

}

class Circle extends Shape{

double radius;

Circle(double x)

{

radius = x;

}

double getArea()

{

return 3.14\*radius\*radius;

}

double getPerimeter()

{

return 6.28\*radius;

}

void draw()

{

System.out.println("Circle");

}

}

class Rectangle extends Shape{

double length;

double breadth;

Rectangle(double x,double y)

{

length = x;

breadth = y;

}

double getArea()

{

return length\*breadth;

}

double getPerimeter()

{

return 2\*(length+breadth);

}

void draw()

{

System.out.println("Rectangle");

}

}

class Square extends Rectangle{

Square(double x)

{

super(x,x);

}

}

class Cylinder extends Circle{

double height;

Cylinder(double x,double y)

{

super(x);

height = y;

}

double getArea()

{

return 2\*3.14\*radius\*(radius + height);

}

double getVolume()

{

return 3.14\*radius\*radius\*height;

}

}

class P6{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int ch=1;

Circle c = new Circle(0);

Rectangle r = new Rectangle(0,0);

Square s = new Square(0);

Cylinder cy = new Cylinder(0,0);

while(ch!=5)

{

System.out.println("1) Circle\n2) Rectangle\n3) Square\n4) Cylinder\n5) Exit");

int chin=1;

ch = sc.nextInt();

switch(ch)

{

case 1: System.out.println("Enter Radius");

double ra = sc.nextDouble();

c = new Circle(ra);

chin = 1;

while(chin!=3)

{

System.out.println("1) Area\n2) Perimeter\n3) Exit");

chin = sc.nextInt();

switch(chin)

{

case 1: System.out.println("Area: " + c.getArea());

break;

case 2: System.out.println("Perimeter: " + c.getPerimeter());

break;

}

}

break;

case 2: System.out.println("Enter Length & Breadth");

double x = sc.nextDouble();

double y = sc.nextDouble();

r = new Rectangle(x,y);

chin = 1;

while(chin!=3)

{

System.out.println("1) Area\n2) Perimeter\n3) Exit");

chin = sc.nextInt();

switch(chin)

{

case 1: System.out.println("Area: " + r.getArea());

break;

case 2: System.out.println("Perimeter: " + r.getPerimeter());

break;

}

}

break;

case 3: System.out.println("Enter Side");

double side = sc.nextDouble();

s = new Square(side);

chin = 1;

while(chin!=3)

{

System.out.println("1) Area\n2) Perimeter\n3) Exit");

chin = sc.nextInt();

switch(chin)

{

case 1: System.out.println("Area: " + s.getArea());

break;

case 2: System.out.println("Perimeter: " + s.getPerimeter());

break;

}

}

break;

case 4: System.out.println("Enter Radius & Height");

double rad = sc.nextDouble();

double height = sc.nextDouble();

cy = new Cylinder(rad,height);

chin = 1;

while(chin!=3)

{

System.out.println("1) Area\n2) Perimeter\n3) Exit");

chin = sc.nextInt();

switch(chin)

{

case 1: System.out.println("Area: " + cy.getArea());

break;

case 2: System.out.println("Perimeter: " + cy.getPerimeter());

break;

}

}

break;

}

}

}

}

Output -

1) Circle

2) Rectangle

3) Square

4) Cylinder

5) Exit

1

Enter Radius

3

1) Area

2) Perimeter

3) Exit

1

Area: 28.259999999999998

1) Area

2) Perimeter

3) Exit

2

Perimeter: 18.84

1) Area

2) Perimeter

3) Exit

3

1) Circle

2) Rectangle

3) Square

4) Cylinder

5) Exit

2

Enter Length & Breadth

2 4

1) Area

2) Perimeter

3) Exit

1

Area: 8.0

1) Area

2) Perimeter

3) Exit

2

Perimeter: 12.0

1) Area

2) Perimeter

3) Exit

3

1) Circle

2) Rectangle

3) Square

4) Cylinder

5) Exit

3

Enter Side

3

1) Area

2) Perimeter

3) Exit

1

Area: 9.0

1) Area

2) Perimeter

3) Exit

2

Perimeter: 12.0

1) Area

2) Perimeter

3) Exit

3

1) Circle

2) Rectangle

3) Square

4) Cylinder

5) Exit

4

Enter Radius & Height

3 6

1) Area

2) Perimeter

3) Exit

1

Area: 169.56

1) Area

2) Perimeter

3) Exit

2

Perimeter: 18.84

1) Area

2) Perimeter

3) Exit

3

1) Circle

2) Rectangle

3) Square

4) Cylinder

5) Exit

5

5) Create an interface **Vehicle**with the methods:

turnIgnitionOn(), turnIgnitionOff(), IncreaseSpeed(int amt), DecreaseSpeed(int amt),  GetSpeed()

Now create a class **Car** that implements this interface.

Create another interface **Flyable** with the following methods:

takeOff(), land()

Now create a class **Helicopter** that will implement both **Vehicle**and **Flyable** interfaces.

**Note** -   
  
In the definition of the functions, you can just print "Taking off..." / "Landing..."

Create a package named **vehicles.**And include the above interfaces and classes in that package. Now import this package and use the classes in a new file.

Program –

// P8.java

import java.util.Scanner;

import vehicle.\*;

class P8

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int ch;

do{

int x;

System.out.println("1) Helicopter\n2) Car");

ch = sc.nextInt();

if(ch == 1)

{

Helicopter h = new Helicopter();

System.out.println("1) Take Off\n2) Land\n3) Turn Ignition On");

System.out.println("4) Turn Ignition Off\n5) Increase Speed\n6) Decrease Speed\n7) Status");

int c = sc.nextInt();

while(c <= 7)

{

switch(c)

{

case 1: h.TakeOff();

break;

case 2: h.Land();

break;

case 3: h.turnignitionOn();

break;

case 4: h.turnIgnitionOff();

break;

case 5: System.out.println("Enter speed");

x = sc.nextInt();

h.IncreaseSpeed(x);

break;

case 6: System.out.println("Enter speed");

x = sc.nextInt();

h.DecreaseSpeed(x);

break;

case 7: h.status();

break;

}

System.out.print(">");

c = sc.nextInt();

}

}

else if(ch == 2)

{

Car h = new Car();

System.out.println("1) Turn Ignition On");

System.out.println("2) Turn Ignition Off\n3) Increase Speed\n4) Decrease Speed\n5) Status");

int c = sc.nextInt();

while(c <= 5)

{

switch(c)

{

case 1: h.turnignitionOn();

break;

case 2: h.turnIgnitionOff();

break;

case 3: System.out.println("Enter speed");

x = sc.nextInt();

h.IncreaseSpeed(x);

break;

case 4: System.out.println("Enter speed");

x = sc.nextInt();

h.DecreaseSpeed(x);

break;

case 5: h.status();

break;

}

System.out.print(">");

c = sc.nextInt();

}

}

}while(ch == 1 || ch == 2);

}

}

// Car.java

package vehicle;

public class Car implements Vehicle

{

private boolean switchon;

private int speed=0;

public void turnignitionOn()

{

switchon = true;

System.out.println("Engine switched on");

}

public void turnIgnitionOff()

{

switchon = false;

speed = 0;

System.out.println("Engine switched off");

}

public void IncreaseSpeed(int amt)

{

speed+= amt;

}

public void DecreaseSpeed(int amt)

{

speed-= amt;

}

public int GetSpeed()

{

return speed;

}

public void status()

{

String s = "The car is ";

if(!switchon)

s = s + "not switched on";

else

s = s + "switched on and moving at " + speed;

System.out.println(s);

}

}

// Flyable.java

package vehicle;

public interface Flyable

{

public void TakeOff();

public void Land();

}

// Helicopter.java

package vehicle;

public class Helicopter implements Vehicle, Flyable

{

private boolean switchon,flying;

private int speed ;

public void turnignitionOn()

{

switchon = true;

System.out.println("Engine switched on");

}

public void turnIgnitionOff()

{

switchon = false;

System.out.println("Engine switched off");

}

public void IncreaseSpeed(int amt)

{

speed+= amt;

}

public void DecreaseSpeed(int amt)

{

speed-= amt;

}

public int GetSpeed()

{

return speed;

}

public void TakeOff()

{

switchon = true;

flying = true;

System.out.println("Taken Off");

}

public void Land()

{

switchon = true;

flying = false;

System.out.println("Landed");

}

public void status()

{

String s = "The helicopter is ";

if(!switchon)

s = s + "not switched on";

else

s = s + "switched on and moving at " + speed;

if(!flying)

s = s + " on land";

else

s = s+ " in the air";

System.out.println(s);

}

}

// Vehicle.java

package vehicle;

public interface Vehicle

{

public void turnignitionOn();

public void turnIgnitionOff();

public void IncreaseSpeed(int amt);

public void DecreaseSpeed(int amt);

public int GetSpeed();

}

Output -

1) Helicopter

2) Car

1

1) Take Off

2) Land

3) Turn Ignition On

4) Turn Ignition Off

5) Increase Speed

6) Decrease Speed

7) Status

1

Taken Off

>2

Landed

>3

Engine switched on

>4

Engine switched off

>5

Enter speed

4

>6

Enter speed

2

>7

The helicopter is not switched on on land

>1

Taken Off

>8

1) Helicopter

2) Car

1

1) Take Off

2) Land

3) Turn Ignition On

4) Turn Ignition Off

5) Increase Speed

6) Decrease Speed

7) Status

1

Taken Off

>5

Enter speed

60

>6

Enter speed

20

>7

The helicopter is switched on and moving at 40 in the air

>2

Landed

>4

Engine switched off

>7

The helicopter is not switched on on land

>8

1) Helicopter

2) Car

2

1) Turn Ignition On

2) Turn Ignition Off

3) Increase Speed

4) Decrease Speed

5) Status

1

Engine switched on

>3

Enter speed

30

>4

Enter speed

10

>5

The car is switched on and moving at 20

>2

Engine switched off

>5

The car is not switched on

>6

1) Helicopter

2) Car

8

6) From your**BankAccount** class (Prev labs), derive two classes**SavingBankAccount, CurrentBankAccount.** Make the appropriate changes in your code. Create a package **Bank**which includes **BankAccount, SavingBankAccount, and CurrentBankAccount**classes.

Program –

// P7.java

import java.util.Scanner;

import Bank.\*;

class P7{

public static void main(String args[])

{

int x = 0;

int ab=0;

float b=0;

char c=0;

String d = "";

Scanner sc = new Scanner( System.in );

SavingsBankAccount S;

CurrentBankAccount B;

while( x == 0 )

{

System.out.println("Enter account no:");

ab = sc.nextInt();

System.out.println("Enter account type:");

c = sc.next().charAt(0);

System.out.println("Enter amount:");

b = sc.nextFloat();

System.out.println("Enter password:");

d = sc.next();

if(b > 1000)

x = 1;

}

if(c == 'c' || c == 'C')

{

B = new CurrentBankAccount(ab,b,d);

while(x == 1)

{

System.out.println("1) Deposit\n2) Withdraw\n3) Check Balance\n4) Display Details\n5) Close Account\n6) Exit");

int a = sc.nextInt();

switch(a)

{

case 1: System.out.println("Enter amount");

B.deposit(sc.nextFloat());

break;

case 2: System.out.println("Enter amount");

B.withdraw(sc.nextFloat());

break;

case 3: System.out.println("The balance is " + B.checkBalance());

break;

case 4: B.disp();

break;

case 5: B.closeAccount();

break;

case 6: System.exit(0);

}

}

}

else

{

S = new SavingsBankAccount(ab,b,d);

while(x == 1)

{

System.out.println("1) Deposit\n2) Withdraw\n3) Check Balance\n4) Display Details\n5) Close Account\n6) Exit");

int a = sc.nextInt();

switch(a)

{

case 1: System.out.println("Enter amount");

S.deposit(sc.nextFloat());

break;

case 2: System.out.println("Enter amount");

S.withdraw(sc.nextFloat());

break;

case 3: System.out.println("The balance is " + S.checkBalance());

break;

case 4: S.disp();

break;

case 5: S.closeAccount();

break;

case 6: System.exit(0);

}

}

}

}

}

//BankAccount.java

package Bank;

import java.util.Scanner;

public class BankAccount{

protected int CustomerId;

protected float Amount;

protected String password;

protected int withdraw;

public BankAccount( int id, float amount, String passcode){

Amount = amount;

CustomerId = id;

password = passcode;

withdraw = 0;

}

public void withdraw(float x){

Scanner st = new Scanner(System.in);

String pass = password;

if(withdraw >= 3)

{

System.out.println("Enter password");

pass = st.next();

if(pass.equals(password))

withdraw = 0;

}

if(Amount - x > 1000 && pass.equals(password))

{

Amount -= x;

System.out.println("Transaction Successful");

withdraw ++;

}

else

{

System.out.println("Transaction Failed");

withdraw ++;

}

}

public void deposit(float x){

withdraw = 0;

Amount += x;

System.out.println("Transaction Successful");

}

public float checkBalance(){

withdraw = 0;

return Amount;

}

public void closeAccount(){

withdraw = 0;

Amount = 0;

}

public void disp(){

withdraw = 0;

System.out.println("ID: " + CustomerId + "\n" + "Amount: " + Amount);

}

}

// CurrentBankAccount.java

package Bank;

public class CurrentBankAccount extends BankAccount{

float limit = 5000;

public CurrentBankAccount( int id, float amount, String passcode)

{

super(id,amount,passcode);

}

public void deposit(float x){

withdraw = 0;

if(x<limit)

{

Amount += x;

System.out.println("Transaction Successful");

}

else

{

System.out.println("Transaction Failed");

}

}

}

//SavingsBankAccount.java

package Bank;

public class SavingsBankAccount extends BankAccount{

float limit = 10000;

public SavingsBankAccount( int id, float amount, String passcode)

{

super(id,amount,passcode);

}

public void deposit(float x){

withdraw = 0;

if(x<limit)

{

Amount += x;

System.out.println("Transaction Successful");

}

else

{

System.out.println("Transaction Failed");

}

}

}

Output –

Output -

Enter account no:

12

Enter account type:

c

Enter amount:

1200

Enter password:

friedchicken

1) Deposit

2) Withdraw

3) Check Balance

4) Display Details

5) Close Account

6) Exit

1

Enter amount

40

Transaction Successful

1) Deposit

2) Withdraw

3) Check Balance

4) Display Details

5) Close Account

6) Exit

2

Enter amount

30

Transaction Successful

1) Deposit

2) Withdraw

3) Check Balance

4) Display Details

5) Close Account

6) Exit

2

Enter amount

30

Transaction Successful

1) Deposit

2) Withdraw

3) Check Balance

4) Display Details

5) Close Account

6) Exit

2

Enter amount

30

Transaction Successful

1) Deposit

2) Withdraw

3) Check Balance

4) Display Details

5) Close Account

6) Exit

2

Enter amount

30

Enter password

friedchicken

Transaction Successful

1) Deposit

2) Withdraw

3) Check Balance

4) Display Details

5) Close Account

6) Exit

3

The balance is 1120.0

1) Deposit

2) Withdraw

3) Check Balance

4) Display Details

5) Close Account

6) Exit

4

ID: 12

Amount: 1120.0

1) Deposit

2) Withdraw

3) Check Balance

4) Display Details

5) Close Account

6) Exit

5

1) Deposit

2) Withdraw

3) Check Balance

4) Display Details

5) Close Account

6) Exit

6