Yash Raj Mani
20BCI0218

DIGITAL ASSIGNMENT

	JAVA

QUESTIONS <1>:

Classes and Objects

Create a Player class to describe a Player details. Add fields such as playerName, country, runScored (Test,T20,one-day). Define default and parametrized constructors to initialize the objects. Get user input and Create atleast 5 objects. Write Java code to display the Objects with runScored in T20 match.

CODE:

```
//CLASS:
  public class player {


    String playername;
    String country;
    int runT20;
    int runTest;
    int runOD;


    //CONSTRUCTOR:
    public player(
      String playername,
      String country,
      int runT20,
      int runTest,
```

```
    int runOD
  ) {
    this.playername = playername;
    this.country = country;
    this.runT20 = runT20;
    this.runTest = runT20;
    this.runOD = runT20;
  }

  public void print() {
    System.out.println("----------------------------");
    System.out.println(" NAME : " + playername);
    System.out.println(" COUNTRY : " + country);
    System.out.println(" RUNS IN T20 : " + runT20);
  }

  public static void main(String[] args) {
    player p1 = new player("YASH", "India", 10, 20, 30);
    player p2 = new player("RAJ", "USA", 50, 40, 60);
    player p3 = new player("RAM", "India", 40, 20, 40);
    player p4 = new player("MOHAN", "China", 60, 70, 60);
    player p5 = new player("RAMESH", "Africa", 90, 60, 70);

    p1.print();
    p2.print();
    p3.print();
```

```
    p4.print();

    p5.print();

  }

}
```

OUTPUT:

```
C:\Users\yashr\Desktop\Java>cd "c:\Users\yashr\Desktop\Jav
-----------------------------
 NAME : YASH
 COUNTRY : India
 RUNS IN T20 : 10
-----------------------------
 NAME : RAJ
 COUNTRY : USA
 RUNS IN T20 : 50
-----------------------------
 NAME : RAM
 COUNTRY : India
 RUNS IN T20 : 40
-----------------------------
 NAME : MOHAN
 COUNTRY : China
 RUNS IN T20 : 60
-----------------------------
 NAME : RAMESH
 COUNTRY : Africa
 RUNS IN T20 : 90

C:\Users\yashr\Desktop\Java>
```

QUESTIONS <2>:

Inheritance

Assume that a bank maintains two kinds of accounts for customers, one called as savings account and the other as current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class account that stores customer name, account number and type of account. From this derive the classes cur_acct and sav_acct to make them more specific to their requirements. Include necessary member functions in order to achieve the following tasks: a) Accept deposit from a customer and update the balance. b) Display the balance c) Compute and deposit interest. d) Permit withdrawal and update the balance. e) Check for the minimum balance, impose penalty, necessary, and update the balance

CODE:

```
import java.lang.*;

import java.util.*;


class Account {


  public String acc_name;

  public double acc_no;
```

```
  public int acc_type;
  public double balance;

  public void getdata(String name, double no, int type, double bal) {
    acc_name = name;
    acc_no = no;
    acc_type = type;
    balance = bal;
  }



}

class Savings extends Account {

  public void deposit(double amt) {
    balance = balance + amt;
    System.out.println(balance);
  }

  public void withdraw(double amt) {
    balance = balance - amt;
    System.out.println(balance);
  }
```

```
  public void interest(int time, int no) {
    double intr = balance * (1 + 6 / no);
    intr = Math.pow(intr, (time * no));
    System.out.println("Intertest :" + intr);
    balance = balance + intr;
    System.out.println("The new balance :" + balance);
  }
}

class Current extends Account {

  public void deposit(double amt) {
    balance = balance + amt;
    System.out.println(balance);
  }
  public void withdraw(double amt) {
    balance = balance - amt;
    System.out.println(balance);
    check(balance);
  }
  public void check(double amt) {
    if (amt < 10000) {
      balance = balance - 500;
```

```
      System.out.println("Insufficient Balance !!!!!!" + balance);
    }
  }
}
public class da {
  public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    int temp = 1;
    while (temp == 1) {
      double amt = 0;
      System.out.println("##############################");

      System.out.println("ENTER NAME :");
      sc.next();
      String name = sc.nextLine();
      System.out.println("ENTER ACC no:");
      double no = sc.nextDouble();
      System.out.println("SELECT\n0 for Savings\n1 for Current");
      int type = sc.nextInt();
      do {
        System.out.println("BALANCE: ");
        amt = sc.nextDouble();
      } while (type == 1 && amt < 10000);
      if (type == 0) {
```

```
        Savings s = new Savings();
        s.getdata(name, no, type, amt);
        System.out.println("##############################");

        System.out.println("\n1.Deposit\n\2.Withdraw\n3.Interest");
        int var = sc.nextInt();
        if (var == 1) {
          System.out.println("Enter Amoumt");
          double amt1 = sc.nextDouble();
          s.deposit(amt1);
        } else if (var == 2) {
          System.out.println("Enter Amoumt");
          double amt1 = sc.nextDouble();
          s.withdraw(amt1);
        } else if (var == 3) {
          System.out.println("Enter time period");
          int tp = sc.nextInt();
          System.out.println("Enter no of times");
          int nof = sc.nextInt();
          s.interest(tp, nof);
        }
      } else if (type == 1) {
        System.out.println("##############################");
```

```
        Current c = new Current();
        c.getdata(name, no, type, amt);
        System.out.println("\n1.Deposit\n\2.Withdraw");
        int var = sc.nextInt();
        if (var == 1) {
          System.out.println("Enter Amoumt");
          double amt1 = sc.nextDouble();
          c.deposit(amt1);
        } else if (var == 2) {
          System.out.println("Enter Amoumt");
          double amt1 = sc.nextDouble();
          c.withdraw(amt1);
        }
      }
      System.out.println("##############################");

      System.out.println("To continue 1 else 0");
      System.out.println("##############################");

      temp = sc.nextInt();

    }
  }
}
```

OUTPUT:

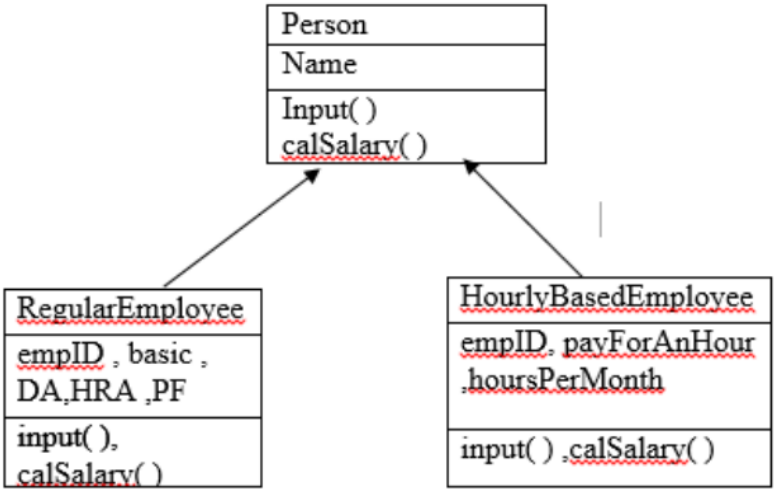
```
C:\Users\yashr\Desktop\Java>cd "c:\Users\yashr\
################################
ENTER NAME :
Yash
ENTER ACC no:
12345
SELECT
0 for Savings
1 for Current
0
BALANCE:
500
################################

1.Deposit
0.Withdraw
SELECT
0 for Savings
1 for Current
0
BALANCE:
200
################################
```

QUESTIONS <3>:

Dynamic Dispatch

Implement the class hierarchy given below.



Use dynamic method dispatch to test the functionality of the above class hierarchy.

CODE:

```
import java.util.*;


class person {


  String name;


  void input() {
    Scanner scan = new Scanner(System.in);
    System.out.println("ENTER THE NAME :");
    name = scan.nextLine();
  }
```

```
  void display() {
    System.out.println("NAME : " + name);
  }
}

class regularemployee extends person {

  String empid;
  double basic, da, hra, pf, salary;

  void input() {
    super.input();
    System.out.println("################################");
    System.out.println("Pls enter regular employee deetails");
    Scanner scan = new Scanner(System.in);
    System.out.println("employee id:");
    empid = scan.nextLine();
    System.out.println("basic pay:");
    basic = scan.nextDouble();
    System.out.println("da:");
    da = scan.nextDouble();
    System.out.println("hra:");
    hra = scan.nextDouble();
    System.out.println("pf:");
    pf = scan.nextDouble();
```

```
    System.out.println("################################");
  }

  void display() {
    super.display();
    salary = basic + da + hra - pf;
    System.out.println("################################");
    System.out.println("regular employee salary:" + salary);
    System.out.println("################################");
  }
}

class hourlybasedemployee {

  String empid;
  double pfa, hpm, salary;

  void input() {
    System.out.println("################################");
    System.out.println("Pls enter hourly based employee details");
    Scanner scan = new Scanner(System.in);
    System.out.println("employee id:");
    empid = scan.nextLine();
    System.out.println("pay for an hour:");
    pfa = scan.nextDouble();
    System.out.println("hours per month:");
```

```
    hpm = scan.nextDouble();
    System.out.println("################################");
  }

  void display() {
    salary = pfa * hpm;
    System.out.println("################################");
    System.out.println("hourly based employee salary" + salary);
    System.out.println("################################");
  }
}

public class Main {

  public static void main(String[] args) {
    regularemployee xx = new regularemployee();
    xx.input();
    regularemployee x;
    x = xx;
    x.display();
    hourlybasedemployee o = new hourlybasedemployee();
    o.input();
    o.display();
  }
}
```

OUTPUT:

```
C:\Users\yashr\Desktop\Java>cd "c:\Users\yashr\Desktop\Java\" && javac Main.java && java Main
ENTER THE NAME :
YASH
################################
Pls enter regular employee deetails
employee id:
1
basic pay:
100
da:
50
hra:
10
pf:
5
################################
NAME : YASH
################################
regular employee salary:155.0
################################
################################
Pls enter hourly based employee details
employee id:
1
pay for an hour:
50
hours per month:
3
################################
################################
hourly based employee salary150.0
################################
```

QUESTIONS <4>:

Packages

Write a program to demonstrate the knowledge of students in working with user-defined packages and sub-packages. Eg., Within the package named ‘primespackage’, define a class Primes which includes a method checkForPrime() for checking if the given number is two digit prime or not. Define another class named TwinPrimes outside of this package which will display all the pairs of prime numbers whose difference is 2. (Eg, within the range 1 to 10, all possible twin prime numbers are (3, 5), (5,7)). The TwinPrimes class should make use of the checkForPrime() method in the Primes class

CODE:

```
package da_pack;


import java.util.Scanner;
import prime_package.Prime;


public class TwinsPrime {


  public static void main(String args[]) {
    int i;


    //take input
    Scanner sc = new Scanner(System.in);
```

```
    System.out.print("Enter 1st : ");
    int n1 = sc.nextInt();
    System.out.print("Enter 2nd : ");
    int n2 = sc.nextInt();
    sc.close();

    System.out.println("TWIN PRIME NUMBERS ARE : ");
    //checking twins prime
    for (i = n1; i <= n2; i++) {
      if (Prime.checkPrime(i) & Prime.checkPrime(i + 2) & i > 1) {
        System.out.println("(" + i + "," + (i + 2) + ")");
      }
    }
  }
}
```

```java
package prime_package;

public class Prime {

    public static boolean checkPrime(int num) {
        int temp;
        boolean isPrime = true;

        for (int i = 2; i <= num / 2; i++) {
            temp = num % i;
            if (temp == 0) {
                isPrime = false;
                break;
            }
        }

        return isPrime;
    }
}
```

OUTPUT:

```
(c) Microsoft Corporation. All rights reserv

C:\Users\yashr\Desktop\da_package> cmd /C ""
:+ShowCodeDetailsInExceptionMessages -cp C:\
dhat.java\jdt_ws\da_package_1499ad10\bin da_
Enter 1st : 20
Enter 2nd : 40
TWIN PRIME NUMBERS ARE :
(29,31)

C:\Users\yashr\Desktop\da_package>
```