JAVA – GrPA Solutions

Week – 3

Implement the code as instructed in the comment, such that it satisfies the given test cases and is in coherence with the given ***main*** method

```
import java.util.*;
class Person{
    private String name;
    private long aadharno;
    public Person(String name, long aadharno){
        this.name = name;
        this.aadharno = aadharno;
    }
    public void print() {
        System.out.println("name : " + name);
        System.out.println("aadharno : " + aadharno);
    }
}

class Employee extends Person{
    private double salary;

    //implement the constructor
public Employee(String nm, long adhano, double salary){
        super(nm, adhano);
        this.salary = salary;
    }
    //override print method
        public void print(){
        super.print();
        System.out.println("salary : " + salary);
    }
}


class ContactEmployee extends Employee{
    final private static double hourlyPay = 100.00;
    private int contactHour;

    //implement the constructor
     public ContactEmployee(String nam, long aadhano, int con){
        super(nam, aadhano, con * hourlyPay);
    }
    //salary is computed as contactHour * hourlyPay
    public void print(){
        super.print();
}
}
```

JAVA – GrPA Solutions

```
44
45  class FClass{
46      public static void main(String[] args) {
47          Scanner sc = new Scanner(System.in);
48          String nm1 = sc.nextLine();
49          String nm2 = sc.nextLine();
50          long adh1 = sc.nextLong();
51          long adh2 = sc.nextLong();
52          double sal = sc.nextDouble();
53          int cont = sc.nextInt();
54          Employee[] eArr = new Employee[2];
55          eArr[0] = new Employee(nm1, adh1, sal);
56          eArr[1] = new ContactEmployee(nm2, adh2, cont);
57          for(Employee e : eArr)
58              e.print();
59      }
60  }
```

JAVA – GrPA Solutions

```java
import java.util.*;
class Shape{
    public int area() {
        return 0;
    }
    public int volume() {
        return 0;
    }
}

class Rectangle extends Shape{
    private int w, h;
//implement Rectangle class
 public Rectangle(int wid, int hei){
        this.w = wid;
        this.h = hei;
    }

    public int area(){
        return w * h;
    }
}

class Cube extends Shape{
    private int a;
    //implement Cube class
     public Cube(int len){
        this.a = len;
    }

    public int volume(){
        return a*a*a;
    }
}
```

JAVA – GrPA Solutions

```
37 class FClass{
38     private static void caller(Shape s) {
39         //check if s is of type Rectangle
40               if (s instanceof Rectangle){
41             System.out.println(s.area());
42         }
43         //check if s is of type Cube
44   if (s instanceof Cube){
45             System.out.println(s.volume());
46         }
47 
48     }
49 public static void main(String[] args) {
50         Scanner sc = new Scanner(System.in);
51         int w = sc.nextInt();
52         int h = sc.nextInt();
53         int a = sc.nextInt();
54         caller(new Rectangle(w, h));
55         caller(new Cube(a));
56     }
57 }
```

JAVA – GrPA Solutions

Create *BankAccount* class that has the following instance variables and methods:

Instance variables:

accountNumber

name

balance

final variable: minBalance

Private method:

checkMinBalance(amount) - returns false if *balance - amount <= minBalance* else returns true

Public methods:

balance( ) - prints the balance

deposit(amount) - updates *balance = balance + amount*

withdraw(amount) - calls the *checkMinBalance(amount)* method,
if it returns true update *balance = balance - amount* else prints Transaction failed

```
import java.util.*;
    class BankAccount{
        int accountNumber;
        String name;
        int balance;
        final int minBalance = 100;
        private boolean checkMinBalance(int amount){
            if(balance - amount <= minBalance){
                return false;
            }
            else{
              return true;
            }
        }
```

```
    //Fill the code here
    public BankAccount(int acc,String n,int bal)
        {
          accountNumber = acc;
          name = n;
          balance = bal;
        }
        public void balance()
        {
          System.out.println(balance);
        }
        public void deposit(int amt)
        {
          balance = balance + amt;
        }
        public void withdraw(int amt)
        {
          if(checkMinBalance(amt) == true)
          {
            balance = balance - amt;
          }
          else
          {
            System.out.println("Transaction failed");
          }
        }
```

```
}
    class AccountCheck{
        public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
            int amnt = sc.nextInt( );
            int amnt1 = sc.nextInt( );
            BankAccount b = new BankAccount(1890, "rahul", 1000);
            b.deposit(amnt);
            b.balance();
            b.withdraw(amnt1);
            b.balance();
        }

    }
```