EXPECTED SOLUTION JUNE 2014

Question 1.1

public void nameBadge()

{

String surname = txfSurname.getText()✓.toUpperCase();✓

Scanner name = new Scanner(txfName.getText());✓

String initials = "";✓

while(name.hasNext())✓

{

initials = initials ✓+ name.next()✓.charAt(0); ✓

}

initials = initials.toUpperCase();

txaOutput.setText("Wakefields Properties\n✓" + surname + " " + initials); ✓

}

**(10)**

public void eligibleToPractice()

{

String id = txfIdNum.getText();

if(id.length()✓!= 13) ✓

{

JOptionPane.showMessageDialog(null , "You have entered an invalid ID number.\nRe-enter the ID number and try again." , - Invalid ID Number" , JOptionPane.ERROR\_MESSAGE);

txfIdNum.requestFocus();

txfIdNum.setText("");

}

else

{

int age = Integer.parseInt(id.substring(0 , 2)); ✓

int year = Calendar.getInstance().getTime().getYear() - 100; ✓

if(age < year) ✓

age = Calendar.getInstance().getTime().getYear() - (age + 100); ✓

else

age = Calendar.getInstance().getTime().getYear() - age; ✓

int gender = id.charAt(6); ✓

boolean eligible = false;

if(gender < 5 ✓&& age >= 21 ✓&& age <= 45) ✓

eligible = true; ✓

else if(gender >= 5 && age >= 21 && age <= 50) ✓

eligible = true;

txaOutput.append("\n\n");

if(eligible)

txaOutput.append("Eligible to Pracice");✓

else

txaOutput.append("Not Eligible “);✓

**(15)**

Question 1.2

public void calculateCommission()

{

try

{

double price = Double.parseDouble✓ (txfPrice.getText())✓;

double commission;

if(rBtnCash.isSelected())✓

{

commission = price \* 7.5 / 100; ✓

txaOuput.setText✓ (String.format("Commission Earned:R%.2f✓" , commission)); ✓

}

else if(rBtnCredit.isSelected())

{

boolean valid = false;

String bondNumber;

do{

bondNumber = JOptionPane.showInputDialog(null , "Enter Bond Number");✓

if(bondNumber.length() != 5) ✓

JOptionPane.showMessageDialog(null , "Bond number must be 5 characters long" , "DDPCoporations - Error" , JOptionPane.ERROR\_MESSAGE);

else if(!Character.isAlphabetic✓ (bondNumber.charAt(0)))

JOptionPane.showMessageDialog(null , "First character of bond number must be a character" , "DDPCoporations - Error" , JOptionPane.ERROR\_MESSAGE);

else if(!Character.isDigit(bondNumber.charAt(4))) ✓

JOptionPane.showMessageDialog(null , "Last character of bond number must be a digit" , "DDPCoporations - Error" , JOptionPane.ERROR\_MESSAGE);

else

valid = true;

}

while(!valid);

commission = price \* 5 / 100; ✓

txaOuput.setText("Bond Number\t" + bondNumber✓ + "\nCommossion Earned:R" + String.format("%.2f" ✓, commission));

}

}

catch(NumberFormatException e)

{

JOptionPane.showMessageDialog(null , "Please enter a valid price" , - Invalid Price Entered" , JOptionPane.ERROR\_MESSAGE);

txfPrice.requestFocus();

txfPrice.setText("");✓

}

} **(15)**

2.1 Employee

public Employee(String name , String duty , int experience , char areaCode)

{

empName = name;

empDuty = duty;

empExperience = experience;

empAreaCode = Character.toUpperCase(areaCode);

}

public double✓ getCommissionPercentage() //can return int

{

switch(empAreaCode) ✓

{

case 'A'✓: return✓ 12.0 / 100; ✓

case 'B': return 7.0 / 100; ✓

case 'C': return 4.0 / 100; ✓

default: return 0; ✓

}

} **(8)**

public double getBonus()

{

return (baseSalary✓ \* empExperience✓ \* 3 / 100✓);

} **(3)**

public double getSalary()

{

double salary = baseSalary;

salary = salary + (salary✓ \* getCommissionPercentage()✓) + getBonus()✓;

return salary;

} **(3)**

public String getEmployeecode()

{

String code = empName.substring(0 , 3); ✓

code = code + empDuty.substring✓ (empDuty.indexOf(“ “)+1) ✓.charAt(0) ✓;

code = code.toUpperCase();✓

return code;

} **(5)**

public String toString() {

String output = "Employee code: " + getEmployeecode()✓;

output = output + "\n✓Employee name✓: " + empName✓;

output = output + "\nArea of duty: " + empDuty;

output = output + "\nYears of experience: " + empExperience;

output = output✓ + String.format("\nSalary: R%.2f✓" , getSalary()✓) + "\n";

return output;

} **(7)**

2.2 Staff Assessment

public StaffAssessment(String agencyName, String category) {

this.agencyName = agencyName;

this.category = category;

} **(1)**

public void setOccupationFlat()

{

occupationFlat = occupationFlat + 1;

} **(1)**

public void setOccupationHouse() {

occupationHouse = occupationHouse = 1;

} **(1)**

public void setOccupationIndustial() {

occupationIndustial = occupationIndustial + 1;

} **(1)**

public void setTotal(double salary) {

totalSalaries = totalSalaries + salary;

} **(1)**

public String toString() {

String output = "Estate Agency: ✓ " + agencyName; ✓

output = output + "\n✓Estate Agents Selling Flats: " + occupationFlat; ✓

output = output + "\nEstate Agents Selling Houses: " + occupationHouse;

output = output✓ +"\nEstate Agents Selling Industrial Parks: " + occupationIndustial; ✓

output = output + "\nTotal Salaries = R" + totalSalaries; ✓

return output;

} **(7)**

}

Quesrtion 2.1

public void process()

{

String name = txfEmpName.getText();

if(name.length() < 3) ✓

{

JOptionPane.showMessageDialog(null , "The employee name must be at least 3 characters long\nPlease re-enter the name and try again" , - Name Too Short" , JOptionPane.WARNING\_MESSAGE); ✓

txfEmpName.requestFocus();

txfEmpName.setText("");

}

else✓

{

String dutyWithSelling = (String)cmbDutyPerformed.getSelectedItem();✓

Scanner sc = new Scanner(dutyWithSelling); ✓

sc.next();

String duty = sc.nextLine().trim();✓

int experience = sldExperience.getValue();✓

char areaCode = ((String)cmbAreaCode.getSelectedItem()).charAt(0); ✓

Employee obj = new Employee(name, duty, experience, areaCode); ✓

txaOutput.setText(obj.toString());✓

}

} **(10)**

Quesrtion 2.2

StaffAssessment obj; ✓

public void ReadFromFile() {

try {

int pos = 1; ✓

Scanner scFile = new Scanner(new File("Assessor.txt"));✓

String firstLine = scFile.nextLine();✓

int per = firstLine.indexOf("%");✓

String name = firstLine.substring(0, per); ✓

String cat = firstLine.substring(per + 1); ✓

obj = new StaffAssessment(name, cat); ✓

txaOuput.append("List of Agents\n");✓

txaOuput.append("===============\n");

while (scFile.hasNext()) {✓

String line = scFile.nextLine();✓

Scanner sc = new Scanner(line).useDelimiter("#");✓

String empName = sc.next();✓

String occType = sc.next();✓

if (occType.equalsIgnoreCase("Houses")) {✓

obj.setOccupationHouse();✓

} else if (occType.equalsIgnoreCase("Flats")) {✓

obj.setOccupationFlat();✓

} else {

obj.setOccupationIndustial();✓

}

double sal = sc.nextDouble();✓

obj.setTotalSalaries(sal); ✓

int r1 = sc.nextInt();

int r2 = sc.nextInt();

int r3 = sc.nextInt();

int r4 = sc.nextInt();✓

int sum = r1 + r2 + r3 + r4; ✓

int score = (int) Math.round((sum / 20.0) \* 100); ✓

txaOuput.append(pos ✓+ ". " + "Name:" + empName + "\n");✓

txaOuput.append(" Occupation: Estate Agent Selling " + occType + "\n");✓

txaOuput.append(" Assessor Score: " + score + "%\n");

if (score > 70) {✓

double bonus = sal \* 0.05; ✓

obj.setTotalSalaries(bonus);

txaOuput.append(" ");

txaOuput.append(String.format("%-10s%-2s%-10.2f\n", "Performance Bonus:", "R", bonus));

} else {

txaOuput.append(" Does Not Receive A Performance Bonus" + "\n\n");✓

}

pos++;

}

scFile.close();

} catch (FileNotFoundException ex) {

JOptionPane.showMessageDialog(null, "File Not Found");

System.exit(0);

}

} **(30)**

public void writeToTextFile() {

try {

PrintWriter fout ✓= new PrintWriter(new FileWriter(obj.getAgencyName()✓ + ".txt"));

fout.append✓ (obj.toString());✓

fout.close();✓

JOptionPane.showMessageDialog(null, "Successfully Written");

} catch (IOException ex) {

JOptionPane.showMessageDialog(null, "Cannot Write to File");

}

} **(5)**

Quesrtion 3

3.1

double[][] arrSales = new double[5][7]; ✓✓ **(2)**

String[] places = {"Ballito" , "La Lucia" , "Umhlanga" , "Izinga Ridge" , "Morning Side"};

3.2

public void generateSales()

{

for (int i = 0; i < 5; i++) {✓

for (int j = 0; j < 6; j++) {✓

arrSales[i][j] ✓ = (Math.random() \* 1700000) + 900000; ✓

}

}

} **(4)**

3.3

public void viewSalesStatistics(){

jTextArea1.setText(String.format("%-15s%-15s%-15s%-15s%-15s%-15s%-15s" , " " , "January" , "February" , "March" , "April" , "May" , "June") + "Total Sales\n");✓

for (int i = 0; i < 5; i++) {✓

jTextArea1.append(String.format("%-15s" , places[i])); ✓

for (int j = 0; j < 7; j++) {✓

jTextArea1.append(String.format("%15.2f" , arrSales[i][j])); ✓

}

jTextArea1.append("\n");✓

}

jTextArea1.append(String.format("%-15s" , "Average"));

for (int i = 0; i < 6; i++) {

jTextArea1.append(String.format("%15.2f" , averages[i])); ✓

}

} **(7)**

3.4

public void swap()

{

for (int i = 0; i < 5; i++) {✓

double temp = arrSales[i][1]; ✓

arrSales[i][1] = arrSales[i][2]; ✓

arrSales[i][2] = temp; ✓

}

} **(4)**

3.5

public void totalSales()

{

for (int i = 0; i < 5; i++) {✓

double sum = 0; ✓

for (int j = 0; j < 6; j++) {✓

sum = sum + arrSales[i][j]; ✓

}

arrSales[i][6] = sum; ✓

}

} **(5)**

3.6

public void averageSales()

{

double[] averages = new double[6];

for (int i = 0; i < 6; i++) {✓

double sum = 0; ✓

for (int j = 0; j < 5; j++) {✓

sum = sum + arrSales[j][i]; ✓

}

averages[i] = sum / 5; ✓

}

} **(5)**

3.7

public void unpopularAreas()

{

jTextArea1.setText("List of Unpopular Areas\n==========================\n");

double lowest = arrSales[0][6];

String place = places[0];

for (int i = 1; i < 5; i++) {✓

if(arrSales[i][6] < lowest) ✓

{

lowest = arrSales[i][6]; ✓

place = places[i]; ✓

}

else if(arrSales[i][6] == lowest)

place = place + "\n" + places[i]; ✓

}

jTextArea1.append(place);

} **(5)**