**Assignment Fulfilment**

Kevin Del Rosario

Diploma in Information Technology, Kaplan Higher Education Institute

IT46 ICS: Introduction to Computer Science

Ms. Vaithilingam Malarkodi

February 9, 2024

**Part 1: Solution Algorithm**

Question 1

1. Defining the Program

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Processing** | **Output** | **Process Control** |
| * sizeInput * weightInput | * set repeat to false to prevent infinite loop * Prompt for sizeInput, weightInput * Get sizeInput, weightInput * Use switch for sizeInput to assign the correct deliveryCharge * Use try-catch to enable repeat when user inputs incorrectly * Assign the deliveryCharge * Display deliveryCharge | * deliveryCharge | * repeat |

1. **Designing a Solution Algorithm (Pseudocode)**
2. Repeat = false
3. do

sizeInput = Read user input for box size

convertedSizeInput = Convert sizeInput to lowercase

3. if sizeInput is not "small" and not "medium" and not "large"

repeat = true

endif

4. try

Display "What is the weight of your parcel in kilograms?"

weightInput = Read user input for parcel weight

5. switch convertedSizeInput

case "small":

if weightInput < 3 and weightInput >= 0

deliveryCharge = 8

repeat = false

else if weightInput >= 3 and weightInput <= 5

deliveryCharge = 12

repeat = false

case "medium":

if weightInput < 5 and weightInput >= 0

deliveryCharge = 15

repeat = false

else if weightInput >= 5 and weightInput <= 10

deliveryCharge = 22

repeat = false

case "large":

if weightInput < 10 and weightInput >= 0

deliveryCharge = 25

repeat = false

else if weightInput >= 10 and weightInput <= 20

deliveryCharge = 35

repeat = false

catch InputMismatchException

repeat = true

while repeat

**Desk Checking the Solution Algorithm**

1. Input Data

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1st Data Set | 2nd Data Set | 3rd Data Set |
|  |  |  |  |
| sizeInput | small | Medium | notbig |
| weightInput | 2 | 4 | 6 |

1. Expected Results

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1st Data Set | 2nd Data Set | 3rd Data Set |
| deliveryCharge | 8 | 15 | Repetition until user inputs a valid input |

1. Desk Check Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Statement No. | sizeInput | weightInput | deliveryCharge | repeat |
| First Pass |  |  |  |  |
| 1 |  |  |  | False |
| 2 | small |  |  |  |
| 3 |  |  |  | False |
| 4 | 2 |  |  |  |
| 5 |  |  | 8 |  |
| Second Pass |  |  |  |  |
| 1 |  |  |  | False |
| 2 | medium |  |  |  |
| 3 |  |  |  | False |
| 4 | 4 |  |  |  |
| 5 |  |  | 15 |  |
| Third Pass |  |  |  |  |
| 1 |  |  |  | False |
| 2 | notbig |  |  |  |
| 3 |  |  |  | True |
| 4 |  | Not reachable unless user inputs correctly |  |  |
| 5 |  |  | Not reachable unless user inputs correctly |  |

Question 2

1. Defining the Program

|  |  |  |
| --- | --- | --- |
| **Input** | **Processing** | **Output** |
| * propertyAnnualValue | * Prompt for propertyAnnualValue * Get propertyAnnualValue * Calculate annualPropertyTax * Display annualPropertyTax | * annualPropertyTax |

1. Designing a Solution Algorithm (Pseudocode)Start

1. Initialize propertyAnnualValue as double

2. Initialize annualPropertyTax as double and set it to 0

3. Try

Output "What is Property's Annual Value?: "

Read propertyAnnualValue from user input

4. If propertyAnnualValue < 0

Output "Invalid Property Value"

Else If propertyAnnualValue < 8000

Set annualPropertyTax to 0

Else If propertyAnnualValue < 55000

Calculate annualPropertyTax as propertyAnnualValue \* 0.04

Else If propertyAnnualValue < 100000

Calculate annualPropertyTax as propertyAnnualValue \* 0.08

Else

Calculate annualPropertyTax as propertyAnnualValue \* 0.12

Output "Your Annual Property Tax is " + annualPropertyTax

Catch InputMismatchException

Output "Wrong Input."

End

1. Desk Checking the Solution Algorithm
   1. Input Data

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1st Data Set | 2nd Data Set | 3rd Data Set |
| propertyAnnualValue | 50000 | 80000 | abc |

* 1. Expected Results

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1st Data Set | 2nd Data Set | 3rd Data Set |
| annualPropertyTax | 2000 | 6400 | Wrong Input |

* 1. Desk Check Table

|  |  |  |
| --- | --- | --- |
| Statement No. | propertyAnnualValue | annualPropertyTax |
| First Pass |  |  |
| 1 |  |  |
| 2 |  | 0 |
| 3 | 50000 |  |
| 4 |  | 2000 |
| Second Pass |  |  |
| 1 |  |  |
| 2 |  | 0 |
| 3 | 80000 |  |
| 4 |  | 6400 |
| Third Pass |  |  |
| 1 |  |  |
| 2 |  | 0 |
| 3 | abc |  |
| 4 |  | Display Wrong Input |

Question 3

1. Defining the Program

|  |  |  |
| --- | --- | --- |
| **Input** | **Processing** | **Output** |
| * employeeID * employeeName * employeeSalary * performanceRating | * read C:\\Users\\User\\OneDrive\\Desktop\\employeefile.txt" * get employeeID, employeeName, employeeSalary, performanceRating * Calculate Bonus,newSalary, description | * id, name, salary, bonus, newSalary, Description |

1. Designing a Solution Algorithm (Pseudocode)

1.Open the file "C:\\Users\\User\\OneDrive\\Desktop\\employeefile.txt" for reading

2.Print column headers: "ID", "Name", "Salary", "Bonus", "New Salary", "Description"

While scanner can still has data to read

3.Read employee details from the file (employeeID, employeeName, employeeSalary, performanceRating)

4. If performanceRating is 1:

Set bonus = employeeSalary \* 0.5

Set revisedSalary = employeeSalary + (employeeSalary \* 0.02)

Set description = "Require Improvement"

Else if performanceRating is 2:

Set bonus = employeeSalary \* 1

Set revisedSalary = employeeSalary + (employeeSalary \* 0.05)

Set description = "Achieved"

Else if performanceRating is 3:

Set bonus = employeeSalary \* 1.5

Set revisedSalary = employeeSalary + (employeeSalary \* 0.08)

Set description = "Exceed"

Else:

Set bonus = 0

Set revisedSalary = 0

Set description = "Invalid Rating"

5. Print employee details (employeeID, employeeName, employeeSalary, bonus, revisedSalary, description)

Close the file

1. Desk Checking the Solution Algorithm
   1. Input Data

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1st Data Set | 2nd Data Set | 3rd Data Set |
| employeeID | 201 | 202 | 203 |
| employeeName | Kevin | Euniese | Len |
| employeeSalary | 2300 | 4500 | 8000 |
| performanceRating | 2 | 3 | 6 |
|  |  |  |  |

* 1. Expected Results

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1st Data Set | 2nd Data Set | 3rd Data Set |
| employeeID | 201 | 202 | 203 |
| employeeName | Kevin | Euniese | Len |
| employeeSalary | 2300 | 4500 | 8000 |
| Bonus | 2300 | 6750 | 0 |
| newSalary | 2415 | 4860 | 0 |
| description | Achieved | Exceed | Invalid Rating |

* 1. Desk Check Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Statement No. | employeeID | employeeName | employeeSalary | Bonus | newSalary | Description |
| First Pass |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |
| 3 | 201 | Kevin | 2300 |  |  |  |
| 4 |  |  |  | 2300 | 2415 | Achieved |
| 5 | Display | Display | Display | Display | Display | display |
| Second Pass |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |
| 3 | 202 | Euniese | 4500 |  |  |  |
| 4 |  |  |  | 6750 | 4860 | Exceed |
| 5 | Display | Display | Display | Display | Display | display |
| Third Pass |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |
| 3 | 203 | Len | 8000 |  |  |  |
| 4 |  |  |  | 0 | 0 | Invalid Rating |
| 5 | Display | Display | Display | Display | Display | display |

**Part 2: Coding**

Question 1:

The program is a delivery charge calculator that prompts users to input their preferred box size (small, medium, or large) and the weight of their parcel in kilograms. The program incorporates a loop to handle invalid inputs and exceptions. The logic employs a switch-case structure based on the converted box size input. Within each case, the program prompts the user for the weight of the parcel and calculates the delivery charge accordingly. The logic includes input validation, ensuring that users provide correct inputs for box size and weight. Additionally, it catches any input mismatch exceptions to handle non-numeric inputs gracefully. The program continues to prompt the user until a valid input is provided, and it displays the calculated delivery charge for the specified box size and weight.

import java.util.\*;

import java.util.InputMismatchException;

public class question1 {

public static void main(String[] args) {

boolean repeat = false;

do {

Scanner sc = new Scanner(System.in);

String sizeInput;

double weightInput;

double deliveryCharge;

System.out.println("Welcome to Delivery Charge Calculator");

System.out.print("What is your Box Size? (Small,Medium, Large): ");

sizeInput = sc.nextLine();

String convertedSizeInput = sizeInput.toLowerCase();

if (!sizeInput.equalsIgnoreCase("small") && !sizeInput.equalsIgnoreCase("medium") && !sizeInput.equalsIgnoreCase("large")) {

System.out.println("Wrong input. Please input correctly.");

repeat = true;

System.out.println("------------------------------------");

}

switch (convertedSizeInput) {

case "small":

try {

System.out.print("What is the weight of your parcel in kilograms?(Lower than or equal to 5): ");

weightInput = sc.nextDouble();

if (weightInput < 3 && weightInput >= 0) {

deliveryCharge = 8;

System.out.println("Delivery Charge: " + deliveryCharge);

repeat = false;

} else if (weightInput >= 3 && weightInput <= 5) {

deliveryCharge = 12;

System.out.println("Delivery Charge: " + deliveryCharge);

repeat = false;

} else {

System.out.println("Your choice of box size is not compatible with the weight.");

repeat = true;

System.out.println("------------------------------------");

}

} catch (InputMismatchException e) {

System.out.println("Wrong input. Please input a valid number.");

repeat = true;

System.out.println("------------------------------------");

}

break;

case "medium":

try {

System.out.print("What is the weight of your parcel in kilograms?(Lower than or equal to 10): ");

weightInput = sc.nextDouble();

if (weightInput < 5 && weightInput >= 0) {

deliveryCharge = 15;

System.out.println("Delivery Charge: " + deliveryCharge);

repeat = false;

}

else if (weightInput >= 5 && weightInput <= 10) {

deliveryCharge = 22;

System.out.println("Delivery Charge: " + deliveryCharge);

repeat = false;

}

else {

System.out.println("Your choice of box size is not compatible with the weight.");

repeat = true;

System.out.println("------------------------------------");

}

} catch (InputMismatchException e) {

System.out.println("Wrong input. Please input a valid number.");

repeat = true;

System.out.println("------------------------------------");

}

break;

case "large":

try {

System.out.print("What is the weight of your parcel in kilograms?(Lower than or equal to 20): ");

weightInput = sc.nextDouble();

if (weightInput < 10) {

deliveryCharge = 25;

System.out.println("Delivery Charge: " + deliveryCharge);

repeat = false;

}

else if (weightInput >= 10 && weightInput <= 20) {

deliveryCharge = 35;

System.out.println("Delivery Charge: " + deliveryCharge);

repeat = false;

}

else {

System.out.println("Your choice of box size is not compatible with the weight.");

repeat = true;

System.out.println("------------------------------------");

}

} catch (InputMismatchException e) {

System.out.println("Wrong input. Please input a valid number.");

repeat = true;

System.out.println("------------------------------------");

}

break;

}

}while (repeat);

}

}

Test Case 1: Valid

A screenshot of a computer

Description automatically generated

Test Case 2: Valid

A screenshot of a computer

Description automatically generated

Test Case 3: Error

A screenshot of a computer

Description automatically generated

Question 2:

The java program calculates the annual property tax based on the user's input for the property's annual value. The program first prompts the user to enter the property's annual value and utilizes a try-catch block to handle potential input mismatch exceptions. If the input is negative, the program informs the user of the invalid property value. The logic then determines the annual property tax based on predefined thresholds for different tax rates. If the annual value is below $8,000, there is no tax; for values between $8,000 and $55,000, the tax is 4%; for values between $55,000 and $100,000, the tax is 8%; and for values exceeding $100,000, the tax is 12%. The program displays the calculated annual property tax to the user.

import java.util.\*;

public class question2 {

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

double propertyAnnualValue, annualPropertyTax = 0;

try {

System.out.print("What is Property's Annual Value?: ");

propertyAnnualValue = sc.nextDouble();

if (propertyAnnualValue < 0) {

System.out.println("Invalid Property Value");

} else if (propertyAnnualValue < 8000) {

annualPropertyTax = 0;

} else if (propertyAnnualValue < 55000) {

annualPropertyTax = propertyAnnualValue \* 0.04;

} else if (propertyAnnualValue < 100000) {

annualPropertyTax = propertyAnnualValue \* 0.08;

} else {

annualPropertyTax = propertyAnnualValue \* 0.12;

}

System.out.println("Your Annual Property Tax is " + annualPropertyTax);

} catch (InputMismatchException e) {

System.out.println("Wrong Input.");

}

}

}

Test Case 1: Valid

A screenshot of a computer

Description automatically generated

Test Case 2: Valid

A screenshot of a computer

Description automatically generated

Test Case 3: Error

A screenshot of a computer

Description automatically generated

Question 3:

It reads employee details from a file, including employee ID, name, salary, and performance rating. It then computes the bonus and the revised salary based on predefined rules for different performance ratings. The program uses a Scanner to read the input file and iterates through each employee's data. Depending on the performance rating, it calculates the bonus and the revised salary with specific multipliers and increments. The results, including employee ID, name, original salary, bonus, revised salary, and a description of the performance, are displayed in tabular format. This program efficiently processes employee information, computes relevant values, and presents the results, providing a clear overview of each employee's performance-related incentives.

import java.util.Scanner;

import java.io.\*;

public class question3 {

public static void main(String[] args) throws Exception{

System.out.println(System.getProperty("user.dir"));

Scanner sc = new Scanner(new File("C:\\Users\\User\\OneDrive\\Desktop\\employeefile.txt"));

String employeeID, employeeName, description;

int performanceRating;

double employeeSalary, bonus, revisedSalary;

System.out.println("ID\t Name\t Salary\t Bonus\t New Salary\t Description");

while (sc.hasNext()) {

employeeID = sc.next();

employeeName = sc.next();

employeeSalary = sc.nextDouble();

performanceRating = sc.nextInt();

if (performanceRating == 1) {

bonus = employeeSalary \* 0.5;

revisedSalary = employeeSalary + (employeeSalary \* 0.02);

description = "Require Improvement";

}

else if (performanceRating == 2) {

bonus = employeeSalary \* 1;

revisedSalary = employeeSalary + (employeeSalary \* 0.05);

description = "Achieved";

}

else if (performanceRating == 3) {

bonus = employeeSalary \* 1.5;

revisedSalary = employeeSalary + (employeeSalary \* 0.08);

description = "Exceed";

}

else {

bonus = 0;

revisedSalary = 0;

description = "Invalid Rating";

}

System.out.println(employeeID + "\t" + employeeName + "\t" +employeeSalary + "\t" +bonus + "\t"+ revisedSalary +"\t\t" + description);

}

}

}

A computer screen shot of a computer screen

Description automatically generated