Intro To Computer Science I and II

Nur Nazifa Sarah Binte Ahmad

CT0368014

PART I :

Question 1

The Next-Day Delivery Service Provider provides the parcel delivery service within Singapore. The delivery charge is calculated based on the parcel weight shown below in the table:

Weight	Charge
Less than 3 kg	8.00
3 - 5 kg	12.00
For each additional 1 kg (Max of 25kg)	1.50

Design an algorithm that will prompt for, receive the parcel weight, calculate the delivery charge and print the value. The program is to repeat the processing until a sentinel weight of 0 is entered.

A. Defining The Program

INPUT	PROCESSING	OUTPUT
- weight	- SET charge = 0, msg to blank - Prompt for weight - Get weight - Calculate delivery_charge - display delivery_charge, msg	- delivery_charge - msg

B. Pseudo Code Algorithm

Delivery_Charges:

```
1. SET delivery_charge=0, msg to blank,
2. Prompt for weight
3. Get weight
4. DOWHILE (weight!=0)
5. IF(weight>=0 AND weight<=3) THEN
                        delivery_charge = 8
                        display delivery_charge
        ELSE
            IF(weight>=3 AND weight<=5) THEN
                        delivery_charge = 12
                        display delivery_charge
          ELSE
                IF(weight>=5 AND weight<=25) THEN
                        delivery_charge = 12 + ((weight−5)*1.5)
                        display delivery_charge
                   ELSE
                        msg = "invalid"
                        display msg
                    ENDIF

               ENDIF

   ENDIF
6. Prompt for weight
7. Get weight
    ENDDO
    END
```

C. Desk Check of Algorithm part I

(i) input data

	First Record	Second Record	Third Record
weight	2	13	-5

C. Desk Check of Algorithm part II

(ii) expected result

	First Record	**Second Record**	**Third Record**
delivery_charge	8	24	-
msg	-	-	Invalid

C. Desk Check of Algorithm part III

(iii) desk check table

Statement Number	**weight**	**delivery_charge**	**msg**	**DOWHILE**
First Pass				
1		0	blank	
2,3	2			
4				TRUE
5		8		
		display		
6,7	13			
4				TRUE
5		24		
		display		
6,7	-5			
4				TRUE
5			invalid	
			display	

Question 2

If you buy a property, you must pay the Buyer's Stamp Duty (BSD). It is calculated by multiplying the purchase price of the property with the stamp duty rate based on either residential or non-residential property as shown below in the table:

Purchase Price of the Property	BSD Rates for Residential Properties	BSD Rates for Non-Residential Properties
Less than $180,000	1%	1.5%
$180,000 - $600,000	2%	2.5%
Above $600,000	3%	3.5%

For example, if the purchase of a non-residential property is $500,000, the BSD would be $500,000 x 2.5% = $12,500. Design an algorithm that will prompt for and receive the purchase price of a property and property type (residential or non-residential) and calculate the buyer's stamp duty and print the value.

A. Defining The Program

INPUT	PROCESSING	OUTPUT
- property - price	- SET bsd = 0, msg to blank - Prompt for property - Get property - Prompt for price - Get price - Calculate bsd - display bsd, msg	- bsd - msg

B. Pseudo Code Algorithm

buyer_stamp_duty:

```
1. SET bsd = 0, msg to blank
2. Prompt for property
3. Get property
4. Prompt for price
5. Get price
6. IF(property == residential) THEN
      IF(price => 0 AND price <= 180,000) THEN
                        bsd = price*0.01
         ELSE IF(price=>180,000 AND price <=600,000) THEN
                        bsd = price*0.02
           ELSE IF(price=> 600,000) THEN
                        bsd = price*0.03
              ELSE
                        msg = "Invalid price!"
              ENDIF
           ENDIF
       ENDIF
   ELSE IF(property == nonResidential) THEN
      IF(price => 0 AND price <= 180,000) THEN
                        bsd = price*0.015
         ELSE IF(price=>180,000 AND price <=600,000) THEN
                        bsd = price*0.025
           ELSE IF(price=> 600,000) THEN
                        bsd = price*0.035
              ELSE
                        msg = "Invalid price!"
              ENDIF
           ENDIF
       ENDIF
    ELSE
      msg = "Invalid property!"
    ENDIF
  ENDIF
7. display bsd
   display msg
   END
```

C. Desk Check of Algorithm part I

(i) input data

	First data set	Second data set	Third data set
property	nonResidential	residential	residential
price	300,000	899,000	-10,000

C. Desk Check of Algorithm part II

(ii) expected result

	First data set	Second data set	Third data set
bsd	7500	26,970	0
msg	-	-	Invalid price!

C. Desk Check of Algorithm part III

(iii) desk check table

Statement Number	price	property	bsd	msg
First pass				
1			0	blank
2,3		nonResidential		
4,5	300000			
6			7500	
7			display	
Second pass				
1			0	blank
2,3		residential		
4,5	899000			
6			26970	
7			display	
Third pass				
1			0	blank
2,3		residential		
4,5	-10000			
6				Invalid price!
7				display

Question 3

There is a mid-year bonus scheme in a sales organisation based on the sales performance for the first half of the year and only payout if the sales performance is on target or more (100% or more). The mid-year bonus is computed using the formula below:

Mid-Year Bonus = Employer's Salary x Sale Performance (%)/ 200

Employees' salary details are maintained in a file that contains employee ID, employee name and employee salary. Design an algorithm that will prompt for and receive the sale performance in percentage and compute the mid-year bonus for each employee in the file and print the employee ID,employee name and bonus.

A. Defining The Program

INPUT	PROCESSING	OUTPUT
- salePerf Employee records - id - name - salary	- Read records - SET bonus = 0, msg to blank - Prompt for salePerf - Get salePerf - Calculate bonus - bonus, msg	- bonus - msg

B. Pseudo Code Algorithm

Mid_Year_Bonus:

1. Read the Employee record
2. DOWHILE NOT EOF
3. SET bonus = 0, msg to blank
4. Prompt for salePerf
5. Get salePerf
6. IF(salePerf > 100%) THEN
bonus = salary * salePerf / 200
display bonus
ELSE
msg = "No Bonus For This Employee!"
display msg
ENDIF
7. Read next record
ENDDO
END

C. Desk Check of Algorithm part I

(i) input data

	First Record	Second Record	Third Record	Fourth Record
id	201	202	203	EOF
name	Jonathan	Marcus	Samantha	
salary	45,000	90,000	60,000	
salePerf	150%	900%	50%	

C. Desk Check of Algorithm part II

(ii) expected result

	First Record	Second Record	Third Record	Fourth Record
bonus	225	4050	-	
msg	-	-	No Bonus For This Employee!	

C. Desk Check of Algorithm part III

(iii) desk check table

Statement Number	**id**	**name**	**salary**	**salePerf** %	**bonus**	**msg**	**DOWHILE**
1	201	Jonathan	45000				
2							TRUE
3					0	blank	
4,5				150			
6					225		
					display		
7	202	Marcus	90000				
2							TRUE
3					0	blank	
4,5				900			
6					4050		
					display		
7	203	Samantha	60000				
2							TRUE
3					0	blank	
4,5				50			
6						No Bonus For This Employee!	
						display	
7	-	-	-				
2							FALSE

PART II :

Implement the three questions algorithm in Part 1 into Java programming. Your code should contain appropriate validations and must focus on code optimisation. You must submit the following:

1. **Three Java code**
2. **Three sample output screenshots**
   - **Two normal and one error test cases**
3. **For each question explain (100 - 150 word) on how the logic works**

(Continue to next page)

Question 1

Java code:

```
import java.util.Scanner;
public class assignmentQ1
{
    public static void main(final String[] args)throws Exception
    {
        Scanner kb = new Scanner(System.in);
        double weight, delivery_charge = 0;
        String msg = "";

        System.out.print("Enter weight : "); //prompt
        weight = kb.nextDouble();

        while(weight != 0)
        {
            //System.out.print("Enter weight value or enter 0 to stop : ");
            //weight = kb.nextDouble();


            if(weight >= 0 && weight <= 3)
            {
                delivery_charge = 8;
                System.out.printf("Delivery charge : %.2f \n", delivery_charge);

            }
            else if(weight >= 3 && weight <=5)
            {
                delivery_charge = 12;
                System.out.printf("Delivery charge : %.2f \n", delivery_charge);

            }
            else if(weight >= 5 && weight <=25)
            {
                delivery_charge = 12 + ((weight-5)*1.5);
                System.out.printf("Delivery charge : %.2f \n", delivery_charge);

            }
            else
            {
                msg = "invalid";
                System.out.println(msg);

            }
            System.out.print("Enter weight or enter 0 to stop : ");
            weight = kb.nextDouble();
        }
        System.out.println("================================");
        System.out.println("Algorithm Terminated!");
    }

}
```

assignmentQ1.java

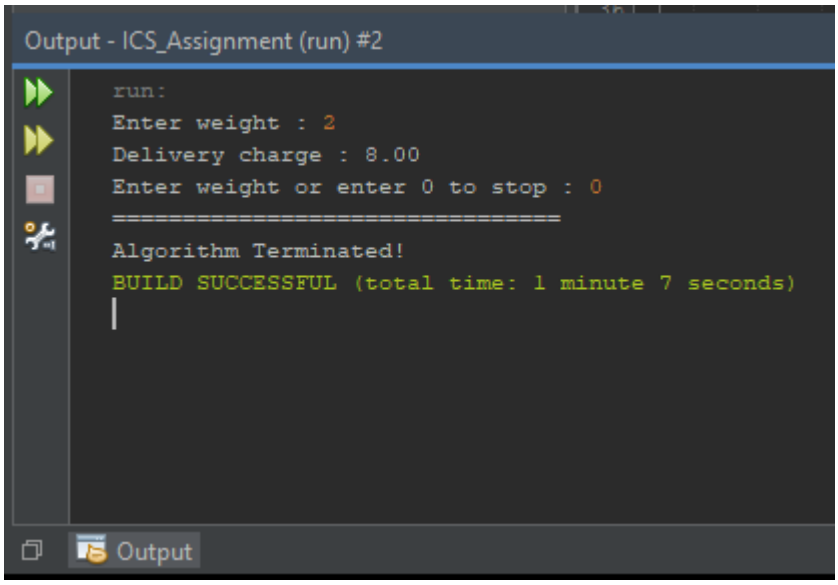
Source History

```java
import java.util.Scanner;
public class assignmentQ1
{
    public static void main(final String[] args)throws Exception
    {
        Scanner kb = new Scanner(System.in);
        double weight, delivery_charge = 0;
        String msg = "";

        System.out.print("Enter weight : "); //prompt
        weight = kb.nextDouble();

        while(weight != 0)
        {
            //System.out.print("Enter weight value or enter 0 to stop : ");
            //weight = kb.nextDouble();


            if(weight >= 0 && weight <= 3)
            {
                delivery_charge = 8;
                System.out.printf("Delivery charge : %.2f \n", delivery_charge);

            }
            else if(weight >= 3 && weight <=5)
            {
                delivery_charge = 12;
                System.out.printf("Delivery charge : %.2f \n", delivery_charge);

            }
            else if(weight >= 5 && weight <=25)
            {
                delivery_charge = 12 + ((weight-5)*1.5);
                System.out.printf("Delivery charge : %.2f \n", delivery_charge);

            }
            else
            {
                msg = "invalid";
                System.out.println(msg);

            }
            System.out.print("Enter weight or enter 0 to stop : ");
            weight = kb.nextDouble();
        }
        System.out.println("================================");
        System.out.println("Algorithm Terminated!");
    }

}
```

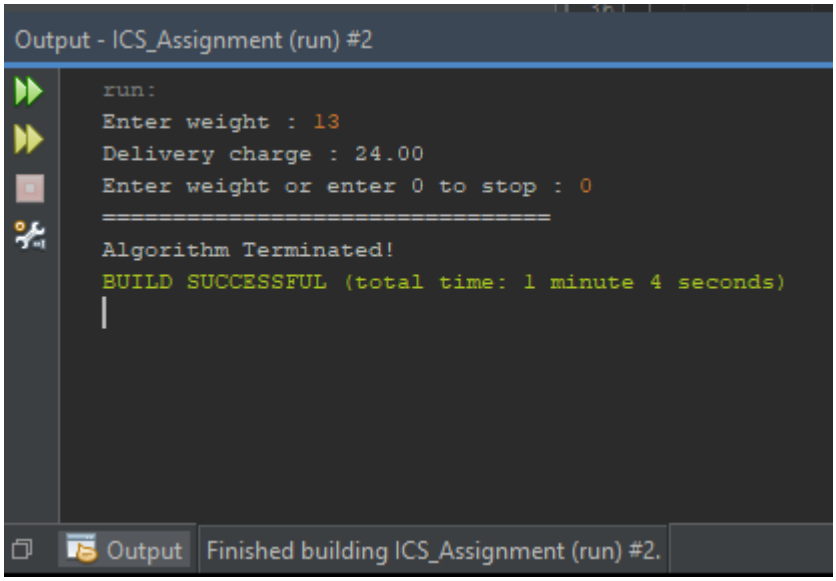
Question 1

Outputs:

1. Valid Data



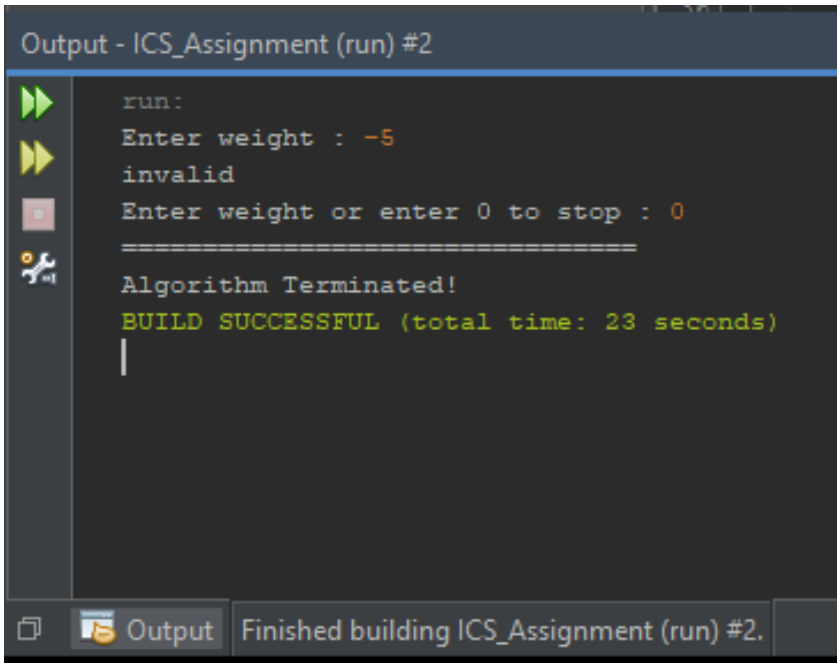
2. Valid Data



Question 1

Outputs:

3. Invalid Data



Logic Brief:

The program for question 1 utilizes the DOWHILE structure in which the condition is tested at the beginning of the loop. The condition stated above uses the not-equal-to operator. If condition is TRUE, it means the number is not equivalent to 0. If so, the algorithm will proceed to the next step. Here, the conditions of the IF statements will then be based between 4 comparisons. The amount charged will be established by the user's weight input after calculation if TRUE. If condition is FALSE, message of invalidity will be displayed. Hence, the delivery charges and message will also be displayed and determined based on the initial input entered by the user for weight. Afterwards, the next execution will go back towards the while condition in order to check if the conditioning is TRUE. If so, the program will repeat the entire algorithmic loop again until a sentinel weight of 0 has been entered by the user.

Question 2

Java code:

```
import java.util.Scanner;
public class assignmentQ2
{
    public static void main(final String[] args)throws Exception
    {
        Scanner kb = new Scanner(System.in);
        double price, bsd = 0;
        String property;
        String residential;
        String nonResidential;
        String msg = "";

        System.out.println("[residential/nonResidential]");
        System.out.println("================================");
        System.out.print("Enter property : "); //prompt
        property = kb.nextLine();
        System.out.println("================================");
        System.out.print("Enter price : "); //prompt
        price = kb.nextInt();
        System.out.println("================================");

        if(property.equalsIgnoreCase("residential"))
        {
            if(price >= 0 && price <= 180000)
            {
                bsd = price * 0.01;
            }
            else if(price >= 180000 && price <= 600000)
            {
                bsd = price * 0.02;
            }
            else if(price > 600000)
            {
                bsd = price * 0.03;
            }
            else
            {
                msg = "Invalid price!";
            }
        }
        else if(property.equalsIgnoreCase("nonResidential"))
        {
            if(price >= 0 && price <= 180000)
            {
                bsd = price * 0.015;
            }
            else if(price >= 0 && price <=600000)
            {
                bsd = price * 0.025;
            }
            else if(price > 600000)
            {
                bsd = price * 0.035;
            }
            else
            {
                msg = "Invalid price!";
            }
        }
        else
        {
            msg = "Invalid property!";
        }

        System.out.printf("Value of the Buyer's Stamp Duty is $%.2f \n", bsd);
        System.out.println("================================");
        System.out.println(msg);
    }
}
```

Q2. Java code (i)

assignmentQ1.java | assignmentQ2.java | assignmentQ3.java

Source | History

```
1
2  import java.util.Scanner;
3  public class assignmentQ2
4  {
5      public static void main(final String[] args)throws Exception
6      {
7          Scanner kb = new Scanner(System.in);
8          double price, bsd = 0;
9          String property;
           String residential;
           String nonResidential;
12         String msg = "";
13
14         System.out.println("[residential/nonResidential]");
15         System.out.println("================================");
16         System.out.print("Enter property : "); //prompt
17         property = kb.nextLine();
18         System.out.println("================================");
19         System.out.print("Enter price : "); //prompt
20         price = kb.nextInt();
21         System.out.println("================================");
22
23         if(property.equalsIgnoreCase("residential"))
24         {
25             if(price >= 0 && price <= 180000)
26             {
27                 bsd = price * 0.01;
28             }
29             else if(price >= 180000 && price <= 600000)
30             {
31                 bsd = price * 0.02;
32             }
33             else if(price > 600000)
34             {
35                 bsd = price * 0.03;
36             }
37             else
38             {
39                 msg = "Invalid price!";
```

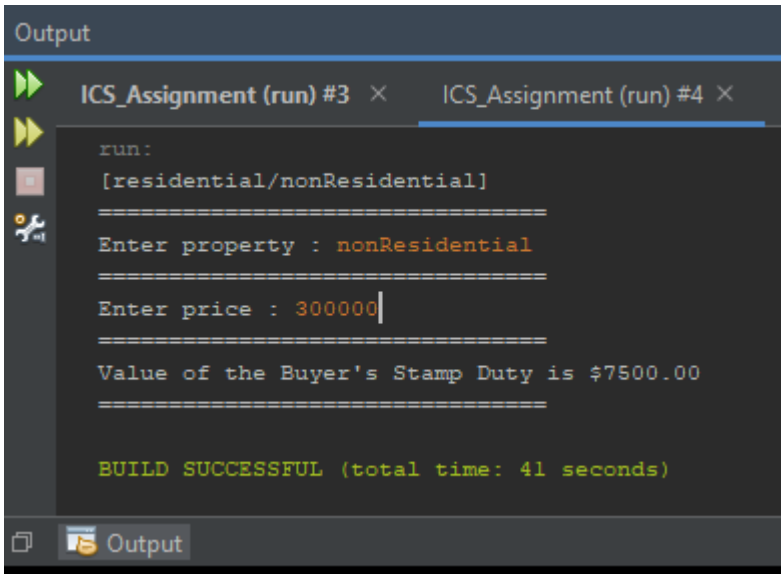
Q2. Java code (ii)

```
            }
        }
        else if(property.equalsIgnoreCase("nonResidential"))
        {
            if(price >= 0 && price <= 180000)
            {
                bsd = price * 0.015;
            }
            else if(price >= 0 && price <=600000)
            {
                bsd = price * 0.025;
            }
            else if(price > 600000)
            {
                bsd = price * 0.035;
            }
            else
            {
                msg = "Invalid price!";
            }
        }
        else
        {
            msg = "Invalid property!";
        }

        System.out.printf("Value of the Buyer's Stamp Duty is $%.2f \n", bsd);
        System.out.println("==============================");
        System.out.println(msg);
    }
}
```

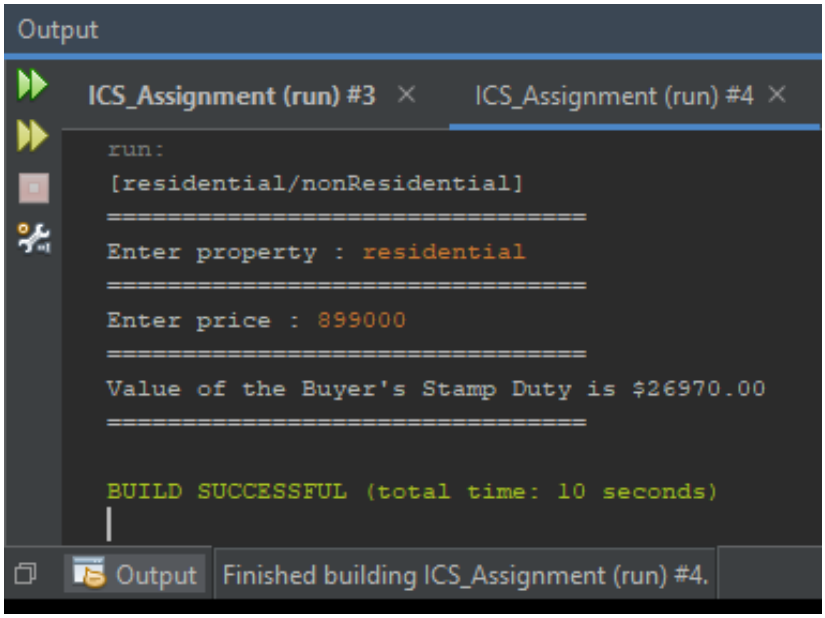
Question 2

Outputs:

1. Valid Data



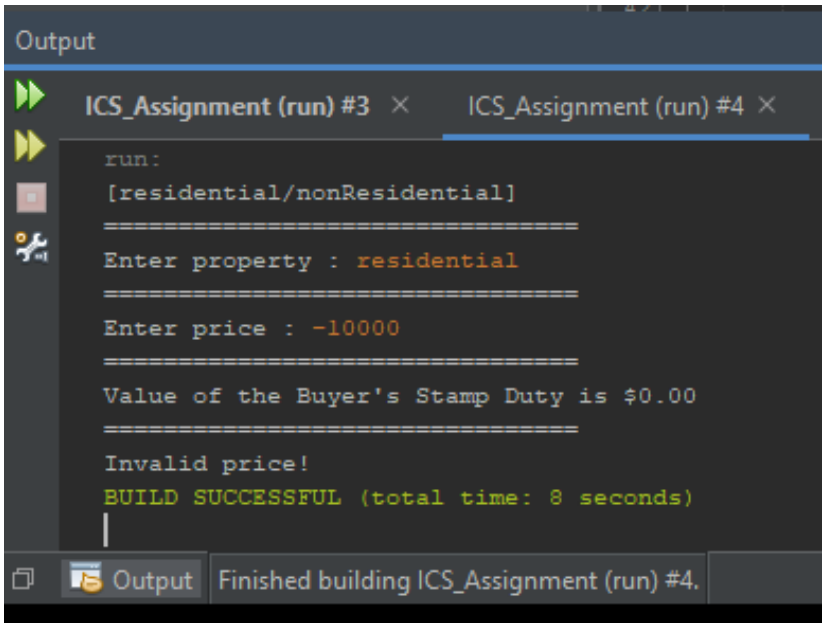
2. Valid Data



Question 2

Outputs:

3. Invalid Data



Logic Brief:

The program for question 2 utilizes the Selection of a non-linear and nested IF statements. If the condition for either properties is TRUE, the algorithm will proceed to the IF statements of either property to satisfy the different conditions before continuance. If condition is FALSE for both properties, the algorithm will proceed to display an invalid message. This will correlate as well if conditions for property is TRUE but is FALSE for the price entered. So, if the conditions for one property and a condition for either of the non-linear nested IFs for pricing is TRUE, algorithm will proceed to calculate and display the price of the Buyer's Stamp Duty.

Question 3

Java code:

```
import java.util.Scanner;
import java.io.*;
public class assignmentQ3
{
    public static void main(final String[] args)throws Exception

    {
        Scanner reader = new Scanner(new File("Employee.txt"));
        Scanner percent = new Scanner(System.in);
        int id, salary, salePerf;
        double bonus;
        String name;
        String msg;



        while(reader.hasNext()) //NOT EOF
        {
            id = reader.nextInt();
            name = reader.next();
            salary = reader.nextInt();
            bonus = 0;
            msg = "";

            System.out.printf("%d \t %10s \t %10d \t %10s \n", id, name, salary, msg);
            System.out.println("====================================");

            System.out.print("Enter sale performance in % : "); //prompt
            salePerf = percent.nextInt();
            System.out.println("====================================");
            if(salePerf > 100)
            {
                bonus = (salary * (salePerf/100)/200);
                System.out.printf("Employee %d's bonus is $%.2f \n", id, bonus);

                System.out.println("====================================");
            }
            else
            {
                msg = "No Bonus For This Employee!";
                System.out.println(msg);
                System.out.println("====================================");

            }

        }

    }
}
```

Q3. Java code (i)

assignmentQ1.java | assignmentQ2.java | assignmentQ3.java

Source | History

```
1  import java.util.Scanner;
2  import java.io.*;
3  public class assignmentQ3
4  {
5      public static void main(final String[] args)throws Exception
6
7      {
8          Scanner reader = new Scanner(new File("Employee.txt"));
9          Scanner percent = new Scanner(System.in);
10         int id, salary, salePerf;
11         double bonus;
12         String name;
13         String msg;
14
15
16
17         while(reader.hasNext()) //NOT EOF
18         {
19             id = reader.nextInt();
20             name = reader.next();
21             salary = reader.nextInt();
22             bonus = 0;
23             msg = "";
24
25             System.out.printf("%d \t %10s \t %10d \t %10s \n", id, name, salary, msg);
26             System.out.println("==================================");
27
28             System.out.print("Enter sale performance in % : "); //prompt
29             salePerf = percent.nextInt();
30             System.out.println("==================================");
31             if(salePerf > 100)
```

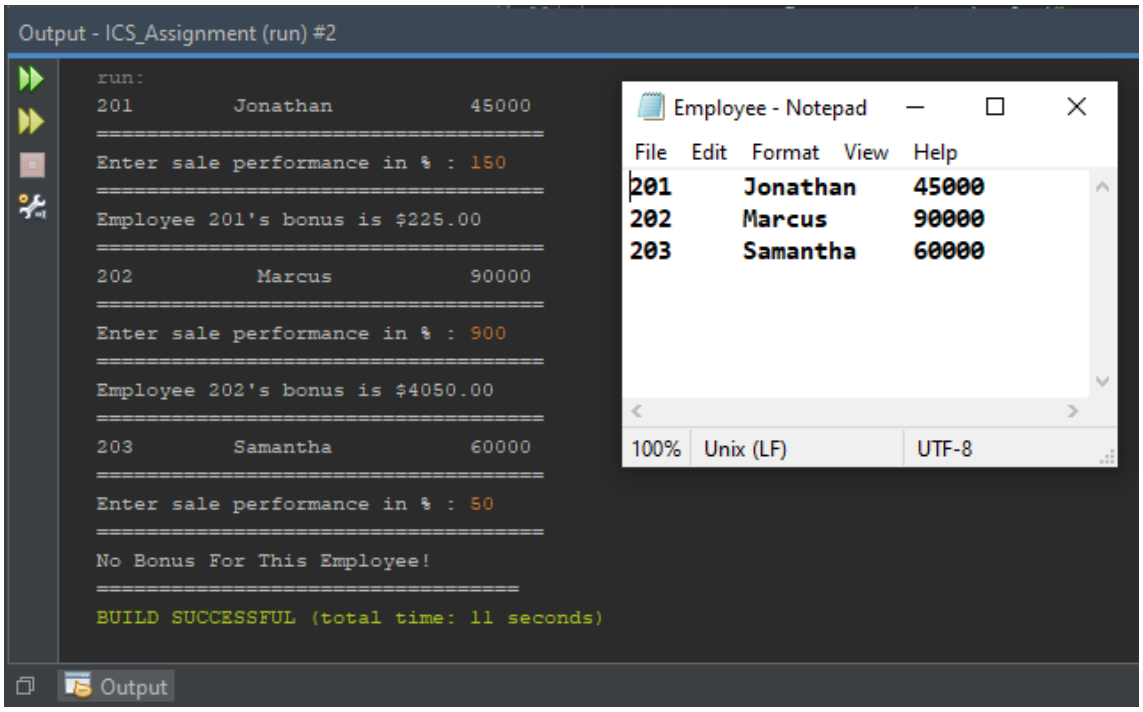
Q3. Java code (ii)

```
32                  {
33                      bonus = (salary * (salePerf/100)/200);
34                      System.out.printf("Employee %d's bonus is $%.2f \n", id, bonus);
35
36                      System.out.println("==================================");
37                  }
38                  else
39                  {
40                      msg = "No Bonus For This Employee!";
41                      System.out.println(msg);
42                      System.out.println("==================================");
43
44                  }
45
46              }
47
48      }
49  }
50
```

Question 3

Outputs:

1, 2 & 3: Valid Data, Valid Data & Invalid Data



Logic Brief:

The program for question 3 utilizes a linear nested IF statement. It is used to test for various values with a different action that is to be taken for each values. The IF statement is also enclosed inside the DOWHILE structure in a counted loop. It will execute a number of times based on the employee records after the program has read the data. If condition is TRUE for the first statement, bonus will then be calculated and displayed. If condition is FALSE, for the initial statement and TRUE for the second statement, a message will be displayed to indicate that no bonus is appointed for the employee. Due to the DOWHILE loop, the program will proceed to read execute the next employee data until completion.