



**UTHM**  
Universiti Tun Hussein Onn Malaysia

**UNIVERSITI TUN HUSSEIN ONN MALAYSIA**

**FINAL EXAMINATION  
SEMESTER II  
SESSION 2018/2019**

COURSE NAME : JAVA PROGRAMMING  
COURSE CODE : BIT 33803  
PROGRAMME : BIT  
EXAMINATION DATE : JUNE / JULY 2019  
DURATION : 3 HOURS  
INSTRUCTION : A) ANSWER ALL QUESTIONS  
B) ANSWER IN THIS QUESTION BOOKLET

THIS QUESTION PAPER CONSISTS OF TEN (10) PAGES

- Q1 (a)** Body Mass Index (BMI) is a measure of health based on height and weight. It can be calculated by taking your weight in kilogram and dividing it by the square of your height in meters. The program in **Figure Q1(a)** prompts a user to enter a weight in kilograms and height in metres. Complete the program in **Figure Q1(a)** to determine Body Mass Index of a user. Display the BMI calculation result with its corresponding interpretation as illustrated in **Table Q1**.

**Table Q1: BMI Interpretation**

BMI	Interpretation
Less than 18.5	Underweight
Greater than or equals to 18.5 and less than 25.0	Normal
Greater than or equals to 25.0 and less than 30.0	Overweight
Greater than or equals to 30.0	Obese

```
import java.util.Scanner;

public class BMI_Interpreter {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // Prompt the user to enter weight
        System.out.print("Enter weight in kg: ");
        double weight = input.nextDouble();

        // Prompt the user to enter height
        System.out.print("Enter height in metres: ");
        double height = input.nextDouble();
        ...
    }
}
```

**Figure Q1(a)**

(10 marks)

**Answer:**

- (b) Rewrite the program in **Figure Q1(b)** using while loop structure to find sum of even values in the range of 1 to 20.

```
class Q1b{
    public static void main(String[] args){
        int sum_even;

        sum_even = 2+4+6+8+10+12+14+16+18+20;

        System.out.println("Sum of even values:" + sum_even);
    }
}
```

**Figure Q1(b)**

(5 marks)

**Answer:**

**Q2 (a)** Determine whether each of the following statements is **TRUE** or **FALSE**.

<b>Statements</b>	<b>Answer</b>
We can define a new class from an existing class. This is known as polymorphism.	
A constructor is used to construct an instance of a class. Like properties and methods, the constructors of a superclass are inherited in the subclass.	
To override a method, the method must be defined in the subclass using the same signature and the same return type as in its superclass.	
A constructor may invoke an overloaded constructor or its superclass's constructor.	
An instance method can be overridden only if it is accessible.	

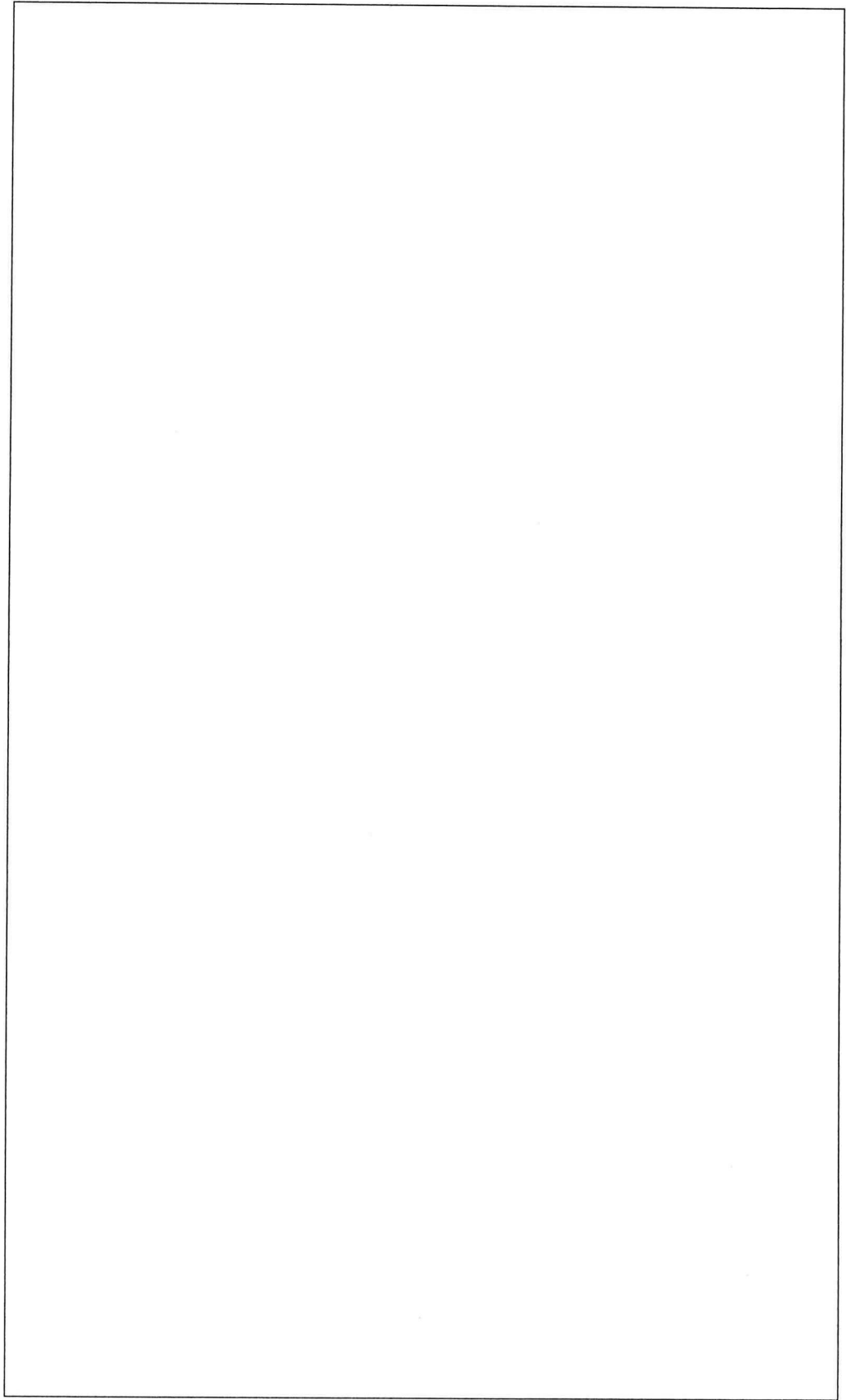
(5 marks)

(b) Draw a UML diagram for a class named `Employee` and implement the class. The class should contain:

- two double data fields named `pay_per_hour` and `number_of_hours_worked` of an employee.
- A no-arg constructor that creates a default employee.
- A constructor that creates an employee with the specified `pay_per_hour` and `number_of_hours_worked`.
- A method named `getTotalPay()` that returns the total salary for an employee.

(20 marks)

**Answer:**



Q3

Based on the specification in **Figure Q3(a)(i)**, complete the GUI program in **Figure Q3(a)(ii)**. The program allows user to enter a length value. Then, it will calculate the conversion of length measure either from feet to centimeters or from centimeters to feet unit.

- Add listeners in the program to handle the events.
- When user clicks the 'Feet to Centimeters' or 'Centimeters to Feet' button, show the result in 'Result' text field.
- The formula for Feet and Centimeters are as follows:

$$\begin{aligned}\text{Feet} &= \text{length} * 30.48 \\ \text{Centimeters} &= \text{length} / 30.4\end{aligned}$$

- The output will be as follows:

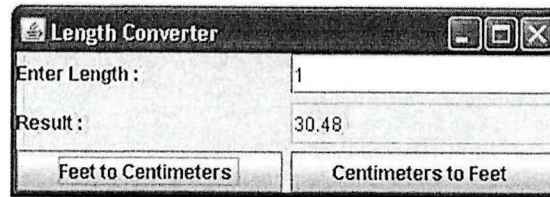


Figure Q3(a)(i)

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

public class FinalExam2 extends JFrame {
    JTextField jtfLength= new JTextField(15);
    JTextField jtfResult = new JTextField(15);

    public FinalExam2() {
        JButton jbtConvert1 = new JButton("Feets to
                                           Centimeters ");
        JButton jbtConvert2 = new JButton("Centimeters to
                                           Feets ");

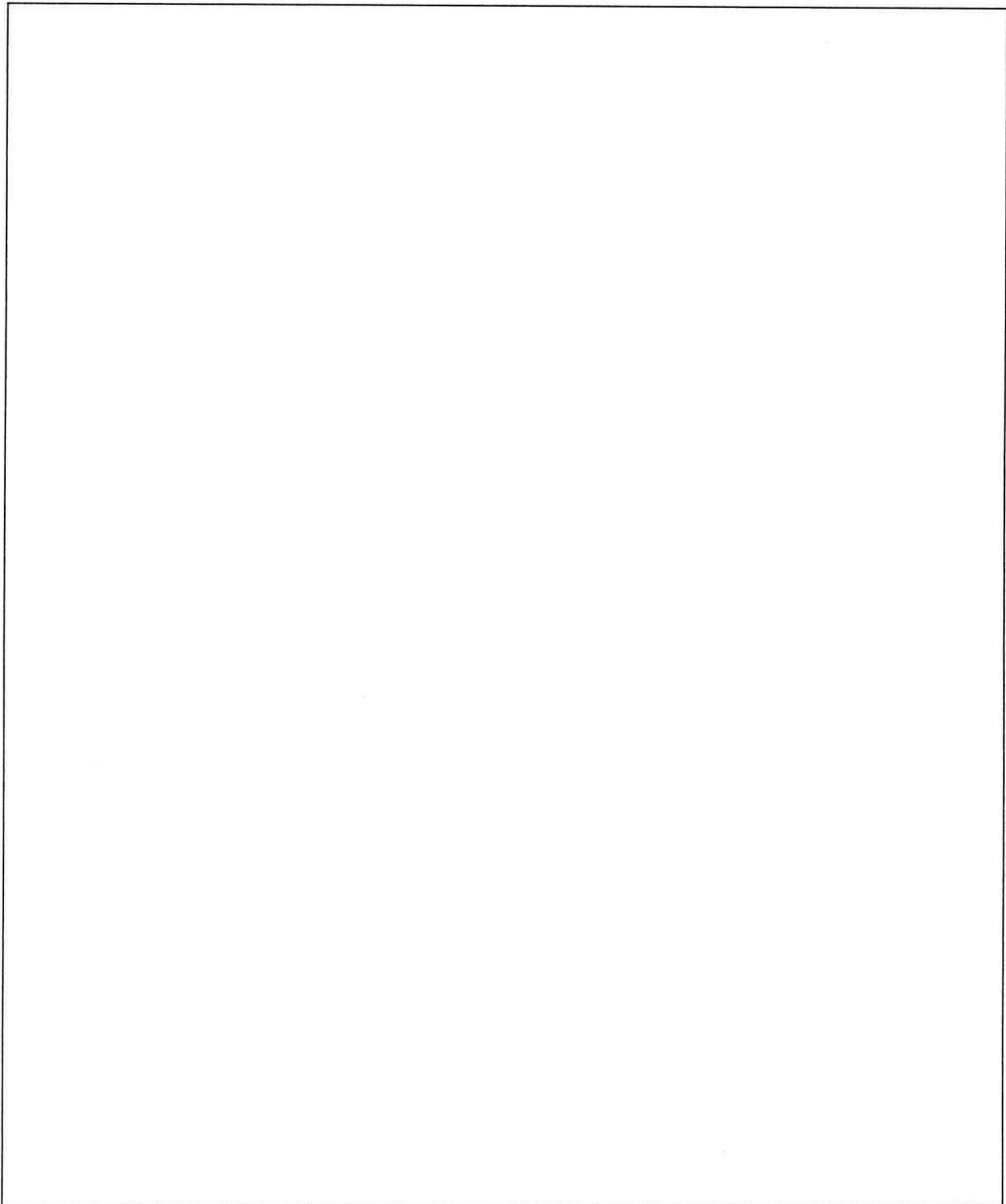
        JPanel p1 = new JPanel();
        jtfResult.setEditable(false);
        p1.setLayout (new GridLayout(3,2,5,5));
        p1.add(new JLabel("Enter Length :"));
        p1.add(jtfLength);
        p1.add(new JLabel ("Result :"));
        p1.add(jtfResult);
        p1.add(jbtConvert1);
        p1.add(jbtConvert2);
        add(p1, FlowLayout.LEFT);
        ...
    }
}
```

```
public static void main(String[] args) {  
    JFrame frame = new FinalExam2();  
    frame.setTitle("Length Converter");  
    frame.setLocationRelativeTo(null); // Center the frame  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    frame.pack();  
    frame.setVisible(true);  
}  
}
```

**Figure Q3(a)(ii)**

(15 marks)

**Answer:**



- Q4 (a)** **Figure Q4(a)** shows a class with a method called `QuotientWithMethod`. The method `quotient` shall return the quotient of two integer values. However, if the second number is 0, the method cannot return a value. Rewrite the program to enable the method to throw an exception that can be caught and handled by the caller.

```
import java.util.Scanner;

public class QuotientWithMethod {
    public static int quotient(int number1, int number2) {
        if (number2 == 0) {
            System.out.println("Divisor cannot be zero");
            System.exit(1);
        }
        return number1 / number2;
    }
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter two integers: ");
        int number1 = input.nextInt();
        int number2 = input.nextInt();
        int result = quotient(number1, number2);
        System.out.println(number1+"/"+number2+"is"+ result);
    }
}
```

**Figure Q4(a)**

(5 marks)

**Answer:**



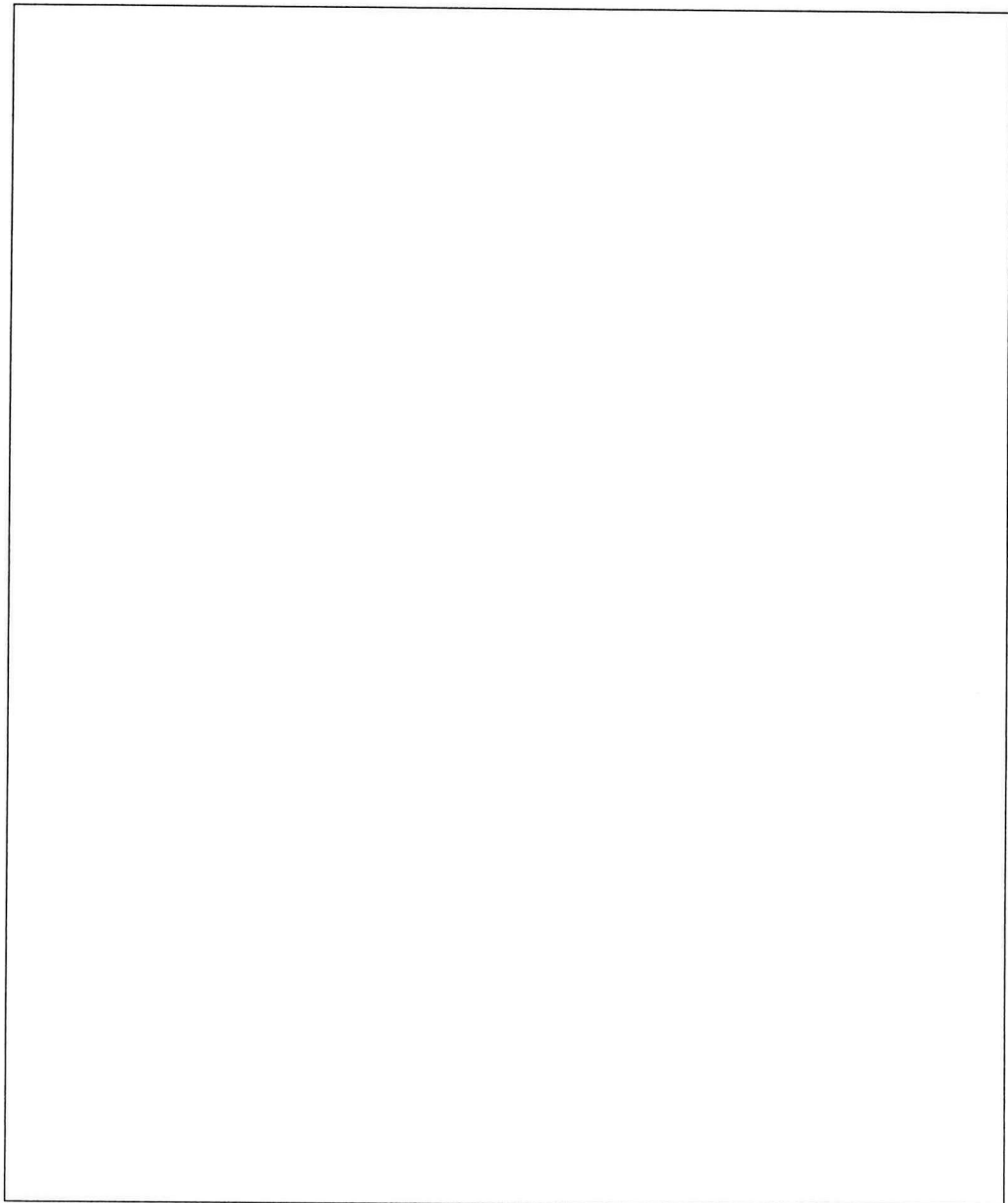
- (b) Write a program that takes in two integer values. The program shall handle exception if a user enter input that is not an integer value. **Figure Q4(b)** illustrates the desired output for the program.

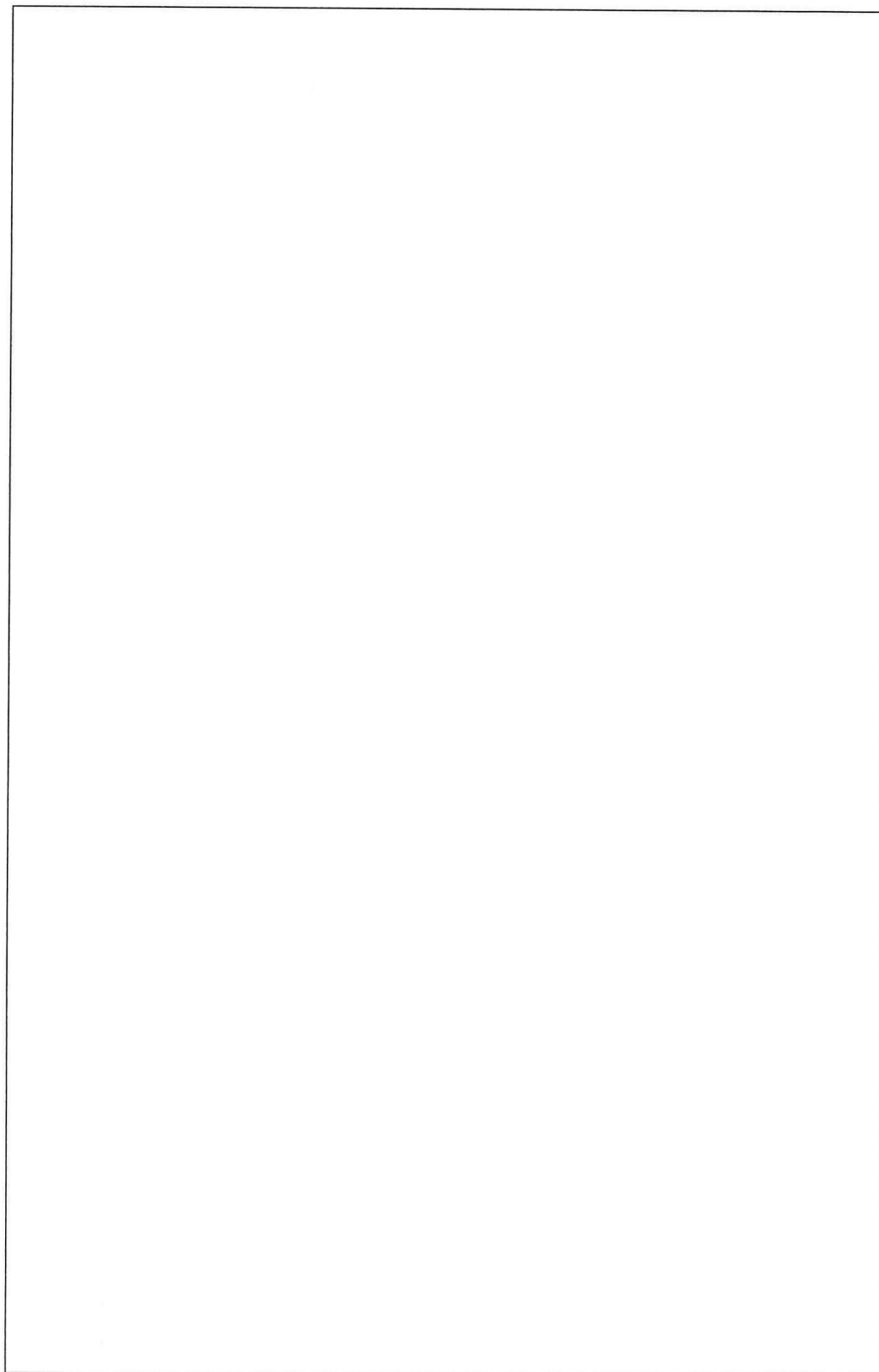
```
Enter an integer: 3.5
Try again. (Incorrect input: an integer is required)
Enter an integer: 4
The number entered is 4
```

**Figure Q4(b)**

(20 marks)

**Answer:**





**- END OF QUESTIONS -**