

University of Petra	 جامعة البترا جامعة البترا	 30 Year Anniversary جامعة البترا - ثلاثون عاما University of Petra
Faculty of Information Technology		كلية تكنولوجيا المعلومات
Department of Computer Science		قسم علم الحاسوب

**Information Technology Fundamentals Lab**

**601106**

**Final Exam – 2025 (1)**

**Form B**

**Your Name:** .....

**Your ID:** .....

**Your Instructor Name:** .....

**Instructions for the Exam:**

- Write your name and ID number on the exam and answer sheets.
- Write the number of the section that you enrolled in.
- Write the name of your instructor.
- Questions in the exam not allowed.
- Using any type of technology (mobiles, smart watches) not allowed
- Using extra papers or sheets not allowed.
- The exam consists of three questions.
- Read the questions from the exam paper and answer on VS Code.
- Compress your answer file and submit the compressed file to Moodle.

**For instructor use only:**

Question number	Course ILO	Program ILO	Question weight	Student mark
<b>Q1</b>	<b>P1</b>		<b>17</b>	
<b>Q2</b>	<b>P2</b>		<b>20</b>	
<b>Q3</b>	<b>P2</b>		<b>8</b>	
			<b>Total /45</b>	

**Question 1:** Write a Java program that meets the following requirements: **(17 Marks)**

- Declare a user-defined method to determine whether an integer is a perfect number. Use the following header:

```
public static boolean isPerfect(int num)
```

A perfect number is a positive integer that is equal to the sum of its proper divisors (excluding itself). For example, `isPerfect(6)` returns true because  $1 + 2 + 3 = 6$ , and `isPerfect(8)` returns false.

- Call the `isPerfect` method from the main method to find the **first 50 perfect numbers** and display every **five perfect numbers in a row**.

**Solution:**

```
public class PerfectNumbers { //1
    public static boolean isPerfect(int num) { //2
        int sum = 0; //0.5
        for (int i = 1; i <= num / 2; i++) { //1.5
            if (num % i == 0) { //2
                sum += i; //0.5
            }
        }
        return sum == num; //1.5
    }
    public static void main(String[] args) { //1
        int count = 0; //0.5
        int number = 1; //0.5
        while (count < 50) { //1
            if (isPerfect(number)) { //1
                System.out.print(number + "\t"); //1
                count++; //1
                if (count % 5 == 0) { //1
                    System.out.println(); //0.5
                }
            }
            number++; //0.5
        }
    }
}
```

**Question 2**

**(20 Marks)**

**Q2.1** Declare a user-defined method **printShape** that prints the following number pattern based on the value of N:

Example when N = 5:

```
1
12
123
1234
12345
```

**Q2.2** Declare another user-defined method named **countDigits(int num)** that accepts one integer and displays **the number of digits** in the given number.

### Main Method Requirements

The main method contains the following:

- Read a positive integer N
- Read an integer num
- Call the methods declared in **Q2.1** and **Q2.2** with the input N and num.

### Solution:

```
import java.util.Scanner; //0.25
public class Test { //0.25
    public static void main(String[] args) { //0.25
        Scanner input = new Scanner(System.in); //0.25
        System.out.print("Enter a positive integer N: "); //0.5
        int N = input.nextInt(); //1
        System.out.print("Enter an integer number: "); //0.5
        int num = input.nextInt(); //1
        PrintShape(N); //1
        countDigits(num); //1
    }
    public static void PrintShape(int n) { //1.5
        for (int i = 1; i <= n; i++) { //1
            for (int j = 1; j <= i; j++) { //1
                System.out.print(j); //0.5
            }
        }
        System.out.println(); //1
    }
}
```

```

    }
}
public static void countDigits(int num) { //1
    int count = 0; //0.5
    int temp = Math.abs(num); //1
    if (temp == 0) { //1
        count = 1; //1
    } else { //0.5
        while (temp > 0) { //1
            count++; //1
            temp /= 10; //1
        }
    }
    System.out.println("Number of digits = " + count); //1
}
}

```

### Question 3

(8 Marks)

Conversion from seconds to minutes and remaining seconds: Write a Java program that displays the following table (note that 1 minute is 60 seconds):

Seconds	Minutes	Remaining Seconds
60	1	0
75	1	15
90	1	30
...		
600	10	0

### Solution:

```

public class SecondsToMinutes { //1
    public static void main(String[] args) { //1
        System.out.println("Seconds\tMinutes\tRemaining Seconds"); //1
        for (int i = 60; i <= 600; i += 15) { //2
            int minutes = i / 60; //1
            int remaining = i % 60; //1
            System.out.println(i + "\t\t" + minutes + "\t\t" + remaining); //1
        }
    }
}
}

```