UNIVERSITY OF MALAYA

*The Leader in Research & Innovation*


# WIX1002 : FUNDAMENTALS OF PROGRAMMING

| | |
|---|---|
| Group name: | Dotech |
| student name : | |
| | Hilya Nadhira (S2191953) |
| | Nurul Amni Faqihah Binti Abdul Halim (22002478) |
| | Syahid Akbar Pasya (22106687) |
| Faculty: | Faculty of Computer Science and Information Technology |
| Title: | Viva1 |
| Semester \| Session: | Sem I 2023/2024 |
| Lecturer: | Dr. Unaizah Obaidellah |

Question 1

| No | Section | Description |
|---|---|---|
| 1 | Problem | Write a program to print out a pattern of an increasing triangle with decreasing column numbers that starts from 0 to 10, with a condition that the outer and inner loop must not be of the same type. |
| 2 | Solution | - Initialize variable n as zero.<br>- Use a while loop as an outer loop with a condition of n is equal or greater than 10 because the number pattern starts with zero and ends with 10.<br>- Use for loop with initialization i equals to n or the number of rows, condition with i is greater and equals with zero, with i incremented<br>- Print i with a space<br>- add n++ to stop the loop |
| 3 | Sample Input & Output | - Viva1 (run)<br>run:<br>0<br>1 0<br>2 1 0<br>3 2 1 0<br>4 3 2 1 0<br>5 4 3 2 1 0<br>6 5 4 3 2 1 0<br>7 6 5 4 3 2 1 0<br>8 7 6 5 4 3 2 1 0<br>9 8 7 6 5 4 3 2 1 0<br>10 9 8 7 6 5 4 3 2 1 0<br>BUILD SUCCESSFUL (total time: 0 seconds) |
| 4 | Source Code | int n=0;<br><br>while(n<=10){<br>for(int i=n; i>=0; i--){<br>System.out.print(i+" ");<br>}<br>System.out.println();<br>n++;<br>} |

Question 2

| No | Section | Description |
|---|---|---|
| 1 | Problem | Write a program that accepts a sentence as an input, counts the number of words in that particular sentence and individually check each word to detect if it is a palindrome and return to final count. With conditions that single words can be counted as palindromes, no leading or trailing spaces exist in the sentence and no punctuation is present. |
| 2 | Solution | - Add a java scanner util.package<br>- Print command for user to enter a sentence<br>- Initialize total words, palindrome, sentence ln, i as zero and check<br>- Enter a while loop<br>- initialize charcheck to check every character in words, and check.length to calculate the length of a string<br>- use a for loop to not count spaces as character<br>- to check if palindrome words, use boolean<br>- enter another while loop to ignore lower or upper case character in words counted as palindrome<br>- Print the total words and palindrome |
| 3 | Sample Input & Output | run:<br>Enter a Sentence: Elle and I like to ride a kayak<br>This Sentence contains 8 words<br>This Sentence contains 4 palindromes<br>BUILD SUCCESSFUL (total time: 8 seconds)<br><br>run:<br>Enter a Sentence:<br>Hannah likes pop music<br>This Sentence contains 4 words<br>This Sentence contains 2 palindromes<br>BUILD SUCCESSFUL (total time: 34 seconds) |
| 4 | Source Code | import java.util.Scanner;<br>public class RandomNumberGenerator {<br>public static void main(String[] args) { |

```
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a Sentence: ");
        String sentence = scanner.nextLine();

        String[] words = sentence.split("\\s+"); // Split the sentence into words
        int totalWords = words.length;
        int palindromeCount = 0;
         for (String word : words) {
            // Remove any non-alphabet characters and convert to lowercase for comparison
            String cleanWord = word.replaceAll("[^a-zA-Z]", "").toLowerCase();
            if (isPalindrome(cleanWord)) {
                palindromeCount++;
            }
        }

        System.out.println("This Sentence contains " + totalWords + " words");
        System.out.println("This Sentence contains " + palindromeCount + " palindromes");
         scanner.close();
    }

    // Function to check if a word is a palindrome
    public static boolean isPalindrome(String word) {
        int left = 0;
        int right = word.length() - 1;
        while (left < right) {
            if (word.charAt(left) != word.charAt(right)) {
                return false;
            }
```

| | | |
|---|---|---|
| | | left++;<br>right--;<br>}<br>return true; |

Question 3

| No | Section | Description |
|---|---|---|
| 1 | Problem | Write a program that generate number 0 to 100 and continue the program with code that runs an infinite loop with a new random number between 0 to 100 is generated in each iteration. The infinite loop should be stopped when the sum total of the random number of 5 times is generated. Printed out the original number and the number of iterations. |
| 2 | Solution | - Generate a random number between 0 and 100 as the original number.<br>- Initialize a counter for the number of iterations.<br>- Enter an infinite loop.<br>- In each iteration, generate a new random number between 0 and 100.<br>- Check if the new number is equal to the original number.<br>- If they are the same, increment a counter for the number of consecutive occurrences. |

<table>
<tr><td></td><td></td><td>- If the consecutive occurrence count reaches 5, print the original number and the total number of iterations, then break out of the loop.<br>- Otherwise, continue to the next iteration.<br>- Finally, display the original number and the total number of iterations it took to generate the same number 5 times.</td></tr>
<tr><td>3</td><td>Sample Input & Output</td><td>run:<br>The number generated is 16<br>It took 611 iterations to generate the number 16 5 times<br>BUILD SUCCESSFUL (total time: 0 seconds)<br><br>run:<br>The number generated is 57<br>It took 367 iterations to generate the number 57 5 times<br>BUILD SUCCESSFUL (total time: 0 seconds)</td></tr>
<tr><td>4</td><td>Source Code</td><td>import java.util.Random;<br><br>public class RandomNumberGenerator {<br>public static void main(String[] args) {<br>Random random = new Random();<br>int originalNumber = random.nextInt(101);<br>System.out.println("The number generated is " + originalNumber);<br><br>int count = 0;<br>int iterations = 0;<br>while (count < 5) {<br>int randomNumber = random.nextInt(101);<br>iterations++;<br>if (randomNumber == originalNumber) {<br>count++;<br>}<br>}</td></tr>
</table>

| | | |
|---|---|---|
| | | System.out.println("It took " + iterations + " iterations to generate the number " + originalNumber + " 5 times");<br>}<br>} |

Question 4

| No | Section | Description |
|---|---|---|
| 1 | Problem | Write a progam which prints a table from 1 to N that contains the number and a column next to it, indicating whether the number is a prime number or not |
| 2 | Solution | - Prompt the user to enter a number N.<br>- Read the input N from the user.<br>- Print a header for the table: "---Table of numbers up to N with their Prime Number status---".<br>- Print the column headers: "Number" and "Prime Number Status".<br>- Use a loop to iterate from 1 to N.<br>- For each iteration, call the checkPrime() method to determine if the current number is prime or not.<br>- Print the current number and its prime number status using printf() or println() with appropriate formatting.<br>- Implement the checkPrime() method to check if a number is prime or not:<br>- a. If the number is less than or equal to 1, return false (not prime).<br>- b. Iterate from 2 to the square root of the number.<br>- c. Check if the number is divisible by any value in this range. If yes, return false (not prime). |

<table>
<tr><td></td><td></td><td>- d. If the loop completes without finding a divisor, return true (prime).<br>- Repeat steps 6-8 for each number in the loop.<br>- End the program.</td></tr>
<tr><td>3</td><td>Sample Input & Output</td><td>

```
run:
Enter a Number: 4
---Table of numbers up to 4 with their Prime Number status---
Number          Prime Number Status
1               1 is a Composite Number
2               2 is a Prime Number
3               3 is a Prime Number
4               4 is a Composite Number
BUILD SUCCESSFUL (total time: 3 seconds)
```

```
run:
Enter a Number: 9
---Table of numbers up to 9 with their Prime Number status---
Number          Prime Number Status
1               1 is a Composite Number
2               2 is a Prime Number
3               3 is a Prime Number
4               4 is a Composite Number
5               5 is a Prime Number
6               6 is a Composite Number
7               7 is a Prime Number
8               8 is a Composite Number
9               9 is a Composite Number
BUILD SUCCESSFUL (total time: 0 seconds)
```

</td></tr>
<tr><td>4</td><td>Source Code</td><td>

```
import java.util.Scanner;


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


        System.out.print("Enter a Number: ");
        int N = scanner.nextInt();


        System.out.println("---Table of numbers up to " + N + "
with their Prime Number status---");
        System.out.println("Number\t\tPrime Number Status");
        for (int i = 1; i <= N; i++) {
            System.out.print(i + "\t\t");
            if (isPrime(i)) {
```

</td></tr>
</table>

```
                System.out.println(i + " is a Prime Number");
            } else {
                System.out.println(i + " is a Composite Number");
            }
        }
    }

    // Helper method to check if a number is prime
    public static boolean isPrime(int number) {
        if (number < 2) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(number); i++) {
            if (number % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```