**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **9 June 2020** | **Name:** | **Shreya poojary** |
| **Course:** | **java** | **USN:** | **4al16ec074** |
| **Topic:** | **udemy** | **Semester & Section:** | **8-B** |
| **Github Repository:** | **Shreya-test** |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Image of session** | | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  | | | |
| **Report** Do While Loop The do-while loop is one of the most common constructs in programming. Versions of it can be found in virtually every programming language. By itself, the loop is fairly straightforward. It tells the computer: “Do these things as long as these conditions are true”.  There is one crucial difference between the Do-While and While loop. In the While loop, the action is performed only if the condition is true. In the above example, if i is greater than 1000, the condition won’t be fulfilled and the output will be blank.  A Do-While loop, on the other hand, will run at least once because the body (Do) is executed before the condition (While) is tested. This can be expressed as follows:  do {  something  }  while (condition)  Thus, the something part is always performed at least once before the program moves on to testing the condition provided under While. do {statement1;statement2; //and so on} while (condition);**Switch Statement in Java** The switch statement is a multi-way branch statement. It provides an easy way to dispatch execution to different parts of code based on the value of the expression. Basically, the expression can be byte, short, char, and int primitive data types. Beginning with JDK7, it also works with enumerated types ( [Enums](https://www.geeksforgeeks.org/enum-in-java/) in java), the [String](https://www.geeksforgeeks.org/string-class-in-java/) class and [Wrapper](https://www.geeksforgeeks.org/primitive-wrapper-classes-are-immutable-in-java/) classes.  **Syntax of Switch-case :**  **// switch statement**  **switch(expression)**  **{**  **// case statements**  **// values must be of same type of expression**  **case value1 :**  **// Statements**  **break; // break is optional**  **case value2 :**  **// Statements**  **break; // break is optional**  **// We can have any number of case statements**  **// below is default statement, used when none of the cases is true.**  **// No break is needed in the default case.**  **default :**  **// Statements**  **}** | | | |
|  | | | |
|  | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |
|  | | | |
|  | | | |
|  | | | |