**DAILY ASSESSMENT FORMAT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **22/06/2020** | **Name:** | **Abhishek Vasudev Mahendrakar** | |
| **Course:** | **C++ Tutorial by SOLOLEARN** | **USN:** | **4AL17EC003** | |
| **Topic:** | 1. **Basic concepts** 2. **Conditionals and Loops** | **Semester & Section:** | **6th-‘A’** | |
| **Github Repository:** | **ECEAbhishekVMahendrakar** | **E-mail:** | **abhi2244mahendrakar@gmail.com** | |
| **FORENOON SESSION DETAILS** | | | |
| **Image of session** | | | |
| **Report – Report can be typed or hand written for up to two pages.**   1. **Basic Concepts:** 2. Welcome to C++: C++ is a general-purpose programming language. C++ is used to create computer programs. Anything from art applications, music players and even video games! C++ was derived from C, and is largely based on it. 3. Your First C++ Program:  * A C++ program is a collection of commands or statements. * Whitespace, such as spaces, tabs, and newlines, is also ignored, although it is used to enhance the program's visual attractiveness. * Program execution begins with the main function, **int main()**. * The next line, **cout << "Hello world!";** results in the display of "Hello world!" to the screen. * In C++, **streams**are used to perform input and output operations. * In most program environments, the standard default output destination is the screen. In C++, **cout**is the stream object used to access it. * **cout**is used in combination with the insertion operator. Write the insertion operator as **<<** to insert the data that comes after it into the stream that comes before. * A **block**is a set of logically connected statements, surrounded by opening and closing curly braces. * The last instruction in the program is the **return**statement. The line **return 0;** terminates the **main()** function and causes it to return the value 0 to the calling process. A non-zero value (usually of 1) signals abnormal termination.  1. Getting the Tools:   You need both of the following components to build C++ programs. 1. **Integrated Development Environment (IDE)**: Provides tools for writing source code. Any text editor can be used as an IDE. 2. **Compiler**: Compiles source code into the final executable program. There are a number of C++ compilers available. The most frequently used and free available compiler is the **GNU C/C++** compiler.  To create a project, open Code::Blocks and click "**Create a new project**" (or File->New->Project). This will open a dialog of project templates. Choose **Console application**and click **Go**.   1. Printing a text:  * You can add multiple insertion operators after **cout**. * The **cout**operator does not insert a line break at the end of the output * One way to print two lines is to use the **endl**manipulator, which will put in a line break. * The new line character **\n** can be used as an alternative to **endl**. * The backslash (\) is called an **escape character**, and indicates a "special" character. * Two newline characters placed together result in a blank line. * Using a single **cout**statement with as many instances of **\n** as your program requires will print out multiple lines of text.  1. Comments:  * **Comments** are explanatory statements that you can include in the C++ code to explain what the code is doing. * The compiler ignores everything that appears in the comment, so none of that information shows in the result. * Comments that require multiple lines begin with **/\*** and end with **\*/**   Within a comment marked with /\* and \*/, // characters have no special meaning, and vice versa. This allows you to "nest" one comment type within the other.   1. Variables:  * Creating a **variable**reserves a memory location, or a space in memory for storing values. The compiler requires that you provide a **data type** for each variable you declare. * Define all variables with a **name**and a**data type** before using them in a program. * In cases in which you have multiple variables of the same type, it's possible to define them in one declaration, separating them with **commas**.  1. Working with Variables:  * To enable the user to input a value, use **cin**in combination with the extraction operator (**>>**). The variable containing the extracted data follows the operator.  1. More on Variables:  * Specifying the data type is required just once, at the time when the variable is declared. * A variable's value may be changed as many times as necessary throughout the program.  1. Basic Arithmetic:      1. Assignment and increment operator:  * The simple **assignment**operator (=) assigns the right side to the left side. * C++ provides shorthand operators that have the capability of performing an operation and an assignment at the same time. * The **increment**operator is used to increase an integer's value by one, and is a commonly used C++ operator. * The increment operator has two forms, **prefix**and **postfix**. * **Prefix**increments the value, and then proceeds with the expression. * **Postfix**evaluates the expression and then performs the incrementing. * The **decrement**operator (--) works in much the same way as the increment operator, but instead of increasing the value, it decreases it by one.  1. **Conditionals and Loops:** 2. **The if else statement:**      * The **if** statement is used to execute some code if a condition is true. * Use **relational operators** to evaluate conditions. * The**not equal to** operator evaluates the operands, determines whether or not they are equal. If the operands are not equal, the condition is evaluated to **true**. * Use relational operators to compare variables in the **if** statement.  1. **The else statement:**  * An **if** statement can be followed by an optional **else**statement, which executes when the condition is **false**. * You can also include, or **nest**, if statements within another if statement. * C++ provides the option of nesting an unlimited number of if/else statements. * In if/else statements, **a single statement** can be included without enclosing it into curly braces.  1. **The while loop:**  * A **loop**repeatedly executes a set of statements until a particular condition is satisfied. * A **while**loop statement repeatedly executes a target statement as long as a given condition remains **true**. * The loop's **body**is the block of statements within curly braces. * The increment value can be changed. If changed, the number of times the loop is run will change, as well.  1. **Using a while loop:**  * The increment or decrement operators can be used to change values in the loop. * A loop can be used to obtain multiple inputs from the user.  1. **The for loop:**  * A **for**loop is a repetition control structure that allows you to efficiently write a loop that executes a specific number of times. * It's possible to change the increment statement.  1. **The do….while loop:**  * A**do...while** loop is similar to a **while**loop. The one difference is that the **do...while** loop is guaranteed to execute **at least one time**. * If the condition evaluated to **false**, the statements in the **do**would still run once. * As with other loops, if the condition in the loop never evaluates to **false**, the loop will run forever.  1. **The switch statement:**  * The **switch**statement tests a variable against a list of values, which are called **cases**, to determine whether it is equal to any of them. * In a switch statement, the optional **default**case can be used to perform a task when none of the cases is determined to be true. * The **break**statement's role is to terminate the switch statement.  1. **Logical operators:**      * Within a single if statement, logical operators can be used to combine **multiple**conditions. * The **OR**(||) operator returns true if any one of its operands is **true**. * The logical **NOT**(!) operator works with just a single operand, reversing its logical state. Thus, if a condition is **true**, the NOT operator makes it **false**, and vice versa. | | | |