**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **23/06/2020** | **Name:** | **Lepakshi T V** |
| **Course:** | **C++** | **USN:** | **4AL17EC044** |
| **Topic:** | * **Data types, arrays and pointers** * **Functions** | **Semester & Section:** | **6th sem A sec** |
| **Github Repository:** | **Lepakshi-044** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report – Report can be typed or hand written for up to two pages.**  The operating system allocates memory and selects what will be stored in the reserved memory based on the variable's **data type**.  **Numeric Data Type:** Numeric data types include: **Integers** (whole numbers), such as -7, 42. **Floating point**numbers, such as 3.14, -42.67.  **Strings & Characters:** A **string**is composed of numbers, characters, or symbols. String literals are placed in **double quotation** marks; some examples are "Hello", "My name is David", and similar. **Characters**are single letters or symbols, and must be enclosed between **single quotes**, like 'a', 'b', etc.  **Booleans:** The Boolean data type returns just two possible values: **true**(1) and **false**(0)  **Integers:** The **integer**type holds non-fractional numbers, which can be positive or negative. Examples of integers would include 42, -42, and similar numbers.  **Floating Point Numbers:** A **floating point** type variable can hold a real number, such as 420.0, -3.33, or 0.03325. The words floating point refer to the fact that a varying number of digits can appear before and after the decimal point. You could say that the decimal has the ability to "**float**". There are three different floating point data types: **float**, **double**, and **long double**.  **Variable Naming Rules:** Use the following rules when naming variables: - All variable names must begin with a letter of the alphabet or an underscore( \_ ). - After the initial letter, variable names can contain additional letters, as well as numbers. Blank spaces or special characters are not allowed in variable names.  **Arrays:** An **array**is used to store a collection of data, but it may be useful to think of an array as a collection of variables that are all of the **same type**.  **Accessing Array Elements:** Index numbers may also be used to assign a new value to an element.  **Multi-Dimensional Arrays:** A **multi-dimensional** array holds one or more arrays. Declare a multidimensional array as follows. type name[size1][size2]...[sizeN];  **Sizeof:**    **Pointers:** Every variable is a **memory**location, which has its **address**defined. That address can be accessed using the **ampersand (&)** operator (also called the address-of operator), which denotes an**address in memory**.  All pointers share the same data type - a long **hexadecimal**number that represents a memory address.  **Static & Dynamic Memory:** To be successful as a C++ programmer, it's essential to have a good understanding of how **dynamic memory** works. In a C++ program, memory is divided into two parts: **The stack**: All of your local variables take up memory from the stack. **The heap**: Unused program memory that can be used when the program runs to **dynamically**allocate the memory.  **Dangling Pointers:** The **delete**operator frees up the memory allocated for the variable, but does not delete the pointer itself, as the pointer is stored on the stack. Pointers that are left pointing to non-existent memory locations are called **dangling pointers**.  **Functions:** A **function**is a group of statements that perform a particular task. You may define your own functions in C++. Using functions can have many advantages, including the following: - You can reuse the code within a function. - You can easily test individual functions. - If it's necessary to make any code modifications, you can make modifications within a single function, without altering the program structure. - You can use the same function for different inputs. #include <iostream> using namespace std;  **Random Numbers:** Being able to generate **random**numbers is helpful in a number of situations, including when creating games, statistical modelling programs, and similar end products. In the C++ standard library, you can access a pseudo random number generator function that's called **rand ()**. When used, we are required to include the header **<cstdlib>**.  **Overloading:** Function **overloading**allows to create multiple functions with the**same name**, so long as they have different parameters.  **Recursion:** A **recursive function** in C++ is a function that calls itself. |