**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **23/06/2020** | **Name:** | **Lavanya B** |
| **Course:** | **C++ programming** | **USN:** | **4al17ec043** |
| **Topic:** | **Data types, Arrays, Pointers**  **Functions** | **Semester & Section:** | **6th A** |
| **Github Repository:** | **Lavanya-B** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report**  **Data types**  **The operating system allocates memory and selects what will be stored in the reserved memory based on the variable's data type.**  **The data type defines the proper use of an identifier, what kind of data can be stored, and which types of operations can be performed.**  **Numeric data types include:**  **Integers (whole numbers), such as -7, 42.**  **Floating point numbers, such as 3.14, -42.67.**  **Strings and 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.**  **Integer**  **The integer type holds non-fractional numbers, which can be positive or negative. Examples of integers would include 42, -42, and similar numbers.**  **Several of the basic types, including integers, can be modified using one or more of these type modifiers:**  **signed: A signed integer can hold both negative and positive numbers.**  **unsigned: An unsigned integer can hold only positive values.**  **short: Half of the default size.**  **long: Twice the default size.**  **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.**  **In most modern architectures, a float is 4 bytes, a double is 8, and a long double can be equivalent to a double (8 bytes), or 16 bytes.**  **Strings**  **A string is an ordered sequence of characters, enclosed in double quotation marks.**  **It is part of the Standard Library.**  **You need to include the <string> library to use the string data type. Alternatively, you can use a library that includes the string library.**  **Eg:**  **#include <string>**  **using namespace std;**  **int main() {**  **string a = "I am learning C++";**  **return 0;**  **}**  **Characters**  **A char variable holds a 1-byte integer. However, instead of interpreting the value of the char as an integer, the value of a char variable is typically interpreted as an ASCII character.**  **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.**  **Instead of declaring multiple variables and storing individual values, you can declare a single array to store all the values.**  **When declaring an array, specify its element types, as well as the number of elements it will hold.**  **Eg:**  **#include <iostream>**  **using namespace std;**  **int main()**  **{**  **int arr[] = {11, 35, 62, 555, 989};**  **int sum = 0;**  **for (int x = 0; x < 5; x++) {**  **sum += arr[x];**  **}**  **cout << sum << endl;**  **return 0;**  **}**  **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.**  **A pointer is a variable, with the address of another variable as its value.**  **In C++, pointers help make certain tasks easier to perform. Other tasks, such as dynamic memory allocation, cannot be performed without using pointers.**  **All pointers share the same data type - a long hexadecimal number that represents a memory address.**  **Eg:**  **#include <iostream>**  **using namespace std;**  **int main()**  **{**  **int score = 5;**  **int \*scorePtr;**  **scorePtr = &score;**  **cout << scorePtr << endl;**  **return 0;**  **}**  **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.**  **Eg:**  **#include <iostream>**  **using namespace std;**  **void printSomething() {**  **cout << "Hi there!";**  **}**  **int main() {**  **printSomething();**  **}**  **Function parameters**  **For a function to use arguments, it must declare formal parameters, which are variables that accept the argument's values. Once parameters have been defined, you can pass the corresponding arguments when the function is called.**  **Eg:**  **#include <iostream>**  **using namespace std;**  **int timesTwo(int x) {**  **return x\*2;**  **}**  **int main() {**  **cout << timesTwo(8) << endl;**  **cout <<timesTwo(5) << endl;**  **cout <<timesTwo(42) << endl;**  **}** |