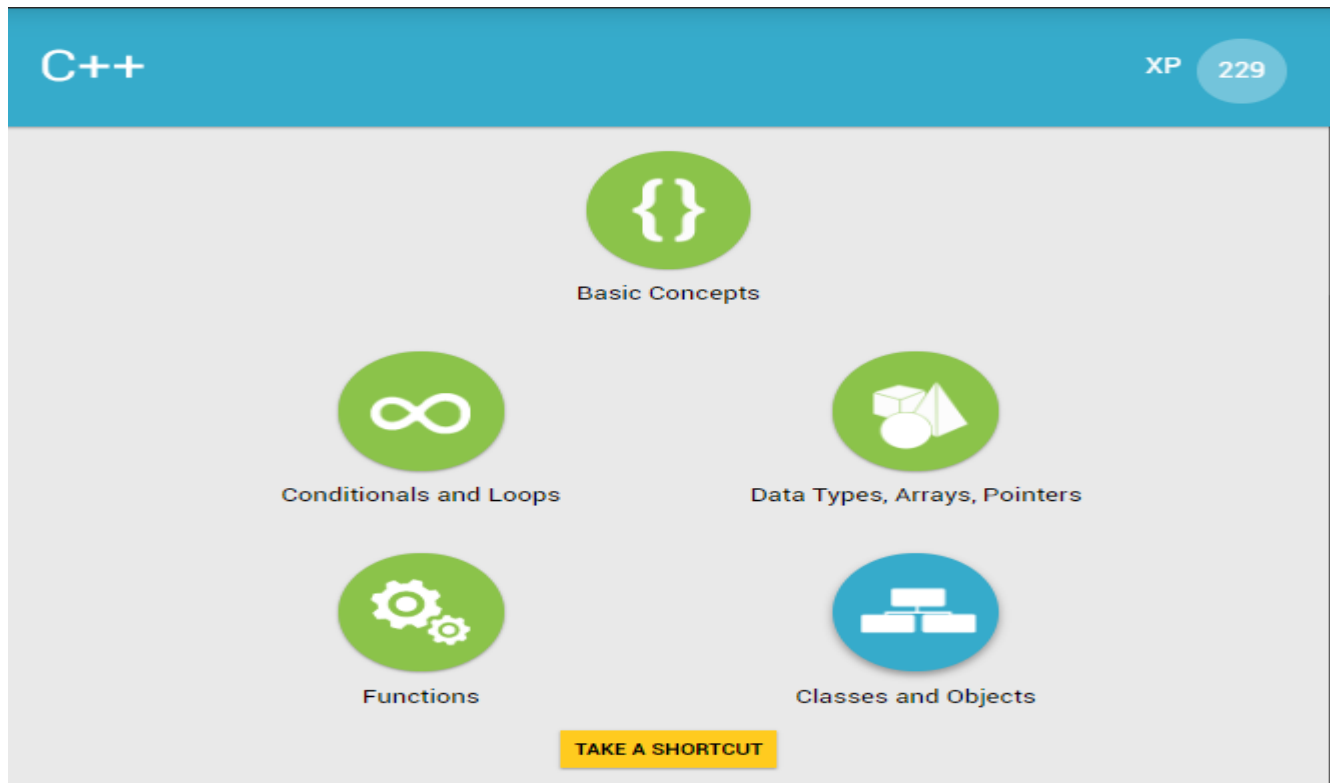


DAILY ASSESSMENT FORMAT

Date:	22/06/2020	Name:	Akshatha M Deshpande
Course:	SoloLearn	USN:	4AL17EC006
Topic:	C++ Programming	Semester & Section:	6th Sem A sec
Github Repository:	AkshathaDeshpande		

FORENOON SESSION DETAILS

Image of session



Report – Report can be typed or hand written for up to two pages.

Module 3:

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:

- 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".

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.
- 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.

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.

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.


Dynamic Memory:

- For local variables on the stack, managing memory is carried out automatically.
- On the heap, it's necessary to manually handle the dynamically allocated memory, and use the delete operator to free up the memory when it's no longer needed.

Sizeof:

- While the size allocated for varying data types depends on the architecture of the computer you use to run your programs, C++ does guarantee a minimum size for the basic data type.

Introduction to Data Types 1/12 5 questions ✓	int, float, double 2/12 3 questions ✓	string, char, bool 3/12 3 questions ✓	Variable Naming Rules 4/12 3 questions ✓
Arrays 5/12 3 questions ✓	Using Arrays in Loops 6/12 3 questions ✓	Arrays in Calculations 7/12 1 questions ✓	Multi-Dimensional Arrays 8/12 3 questions ✓
Introduction to Pointers 9/12 4 questions ✓	More on Pointers 10/12 2 questions ✓	Dynamic Memory 11/12 5 questions ✓	The sizeof() Operator 12/12 2 questions ✓
Module 3 Quiz 8 questions ✓			


Akshatha M Deshpande
akshathadeshpande1103@gmail.com
Reset | Sign out

MODULE 2:

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.
- **return-type:** Data type of the value returned by the function.
- **function name:** Name of the function.
- **parameters:** When a function is invoked, you pass a value to the parameter. This

value is referred to as actual parameter or argument. The parameter list refers to the type, order, and number of the parameters of a function.

- **body of the function:** A collection of statements defining what the function does.

Multiple Parameters:

- You can define as many parameters as you want for your functions, by separating them with commas.

Random Numbers:

- Being able to generate random numbers is helpful in a number of situations, including when creating games, statistical modeling 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>`.

The `srand()` Function:

- The `srand()` function is used to generate truly random numbers.
- This function allows to specify a seed value as its parameter, which is used for the `rand()` function's algorithm.

Default Values for Parameters:

- When defining a function, you can specify a default value for each of the last parameters. If the corresponding argument is missing when you call a function, it uses the default value.
- To do this, use the assignment operator to assign values to the arguments in the function definition, as shown in this example.

Overloading:

- Function overloading allows to create multiple functions with the same name, so long as they have different parameters.

Function Overloading:

- When overloading functions, the definition of the function must differ from each other by the types and/or the number of arguments in the argument list.

Recursion:

- A recursive function in C++ is a function that calls itself.
- To demonstrate recursion, let's create a program to calculate a number's factorial.
- In mathematics, the term factorial refers to the product of all positive integers that are less than or equal to a specific non-negative integer (n).
- The factorial of n is denoted as $n!$

Arrays and Functions:


- An array can also be passed to a function as an argument.
- The parameter should be defined as an array using square brackets, when

declaring the function.

Function Arguments:

- There are two ways to pass arguments to a function as the function is being called.
- **By value:** This method copies the argument's actual value into the function's formal parameter. Here, we can make changes to the parameter within the function without having any effect on the argument.
- **By reference:** This method copies the argument's reference into the formal parameter. Within the function, the reference is used to access the actual argument used in the call. This means that any change made to the parameter affects the argument.

Introduction to Functions 1/9 6 questions ✓	Function Parameters 2/9 3 questions ✓	Functions with Multiple Parameters 3/9 4 questions ✓	The rand() Function 4/9 5 questions ✓
Default Arguments 5/9 2 questions ✓	Function Overloading 6/9 3 questions ✓	Recursion 7/9 4 questions ✓	Passing Arrays to Functions 8/9 2 questions ✓
Pass by Reference with Pointers 9/9 4 questions ✓	Module 4 Quiz 6 questions ✓		



Akshatha M Deshpande
akshathadeshpande1103@gmail.com
[Reset](#) | [Sign out](#)