

DAILY ASSESSMENT FORMAT

Date:	23rd June 2020	Name:	Sushmitha R Naik
Course:	C++ programming	USN:	4AL17EC090
Topic:	Data types, Arrays, pointer	Semester & Section:	6th sem 'B' sec
GitHub Repository:	Sushmitha_naik		

FORENOON SESSION DETAILS

Image of session

The screenshot displays the SOLOLEARN interface for the 'Data Types, Arrays, Pointers' course. The header shows the course title and a progress bar with 146 XP. The main content area is a grid of topics and quizzes, each with a green checkmark indicating completion. The topics are arranged in a 4x4 grid, with the last cell containing a 'Module 3 Quiz'.

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

On the right side of the interface, the user's profile is shown: SUSHMITHA NAIK, with a link to her email (sushmithanaik456@gmail.com) and options to 'Reset' or 'Sign out'. The footer indicates '© 2020 SoloLearn Inc.'.

Data Types:

A data type specifies the type of data that a variable can store such as integer, floating, character etc. There are 4 types of data types in C++ language.

1. Basic Data Type- int, char, float, double, etc
2. Derived data type- arrays, pointer etc
3. Enumeration Data Type- enum
4. User defined data type- structure

Basic Data Types:

The basic data types are integer-based and floating-point based. C++ language supports both signed and unsigned literals. The memory size of basic data types may change according to 32 or 64 bit operating system.

Arrays:

Like other programming languages, array in C++ is a group of similar types of elements that have contiguous memory location.

In C++ `std::array` is a container that encapsulates fixed size arrays. In C++, array index starts from 0. We can store only fixed set of elements in C++ array.

Advantages of C++ Array

- Code Optimization (less code)
- Random Access
- Easy to traverse data
- Easy to manipulate data
- Easy to sort data etc.

Disadvantages of C++ Array

- Fixed size

Array Types:

There are 2 types of arrays in C++ programming:

1. Single Dimensional Array
2. Multidimensional Array

Single Dimensional Array:

Let's see a simple example of C++ array, where we are going to create, initialize and traverse array. #include

```
<iostream>
using namespace std;
int main()
{
    int arr[5]={10, 0, 20, 0, 30}; //creating and initializing array
    //traversing array
    for (int i = 0; i < 5; i++)
    {
        cout<<arr[i]<<"\n";
    }
}
```

Multidimensional Arrays:

The multidimensional array is also known as rectangular arrays in C++. It can be two dimensional or three dimensional. The data is stored in tabular form (row * column) which is also known as matrix.

```
#include <iostream>
using namespace std;
int main()
{
    int test[3][3]; //declaration of 2D array
    test[0][0]=5; //initialization test[0][1]=10;
    test[1][1]=15;
    test[1][2]=20;
    test[2][0]=30;
    test[2][2]=10;
    //traversal
    for(int i = 0; i < 3; ++i)
    {
        for(int j = 0; j < 3; ++j)
        {
            cout<< test[i][j]<<" ";
        }
        cout<<"\n"; //new line at each row
    }
    return 0;
}
```

Pointers:

The pointer in C++ language is a variable, it is also known as locator or indicator that points to an address of a value.

Advantage of pointer

1) Pointer reduces the code and improves the performance, it is used to retrieving strings, trees etc. and used with arrays, structures and functions.

We can return multiple values from function using pointer.

3) It makes you able to access any memory location in the computer's memory.

Usage of pointer

There are many usage of pointers in C++ language.

1) Dynamic memory allocation

In c language, we can dynamically allocate memory using malloc() and calloc() functions where pointer is used.

2) Arrays, Functions and Structures

Pointers in c language are widely used in arrays, functions and structures. It reduces the code and improves the performance.

Pointer Program to swap 2 numbers without using 3rd variable

```
#include <iostream>
using namespace std;
int main()
{
    int a=20,b=10,*p1=&a,*p2=&b;
    cout<<"Before swap: *p1="<<*p1<<" *p2="<<*p2<<endl;
    *p1=*p1+*p2;
    *p2=*p1-*p2;
    *p1=*p1-*p2;
    cout<<"After swap: *p1="<<*p1<<" *p2="<<*p2<<endl;
    return 0;
}
```

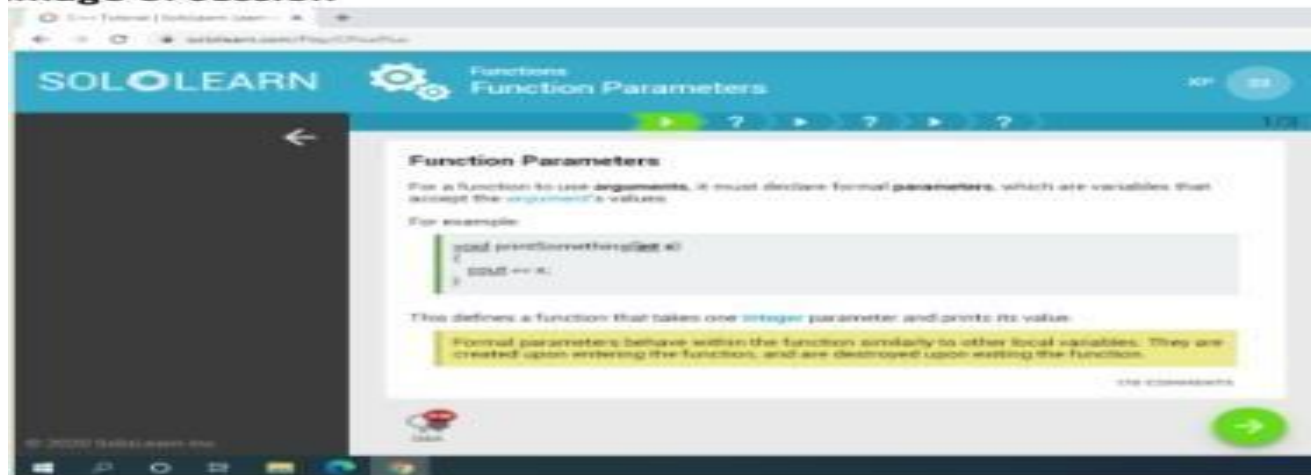
DAILY ASSESSMENT FORMAT

Date:	23 rd June 2020	Name:	Sushmitha_naik
Course:	C++ programming	USN:	4AL17EC090
Topic:	Functions	Semester & Section:	6 th sem 'B' sec
GitHub Repository:	Sushmitha_naik		

AFTERNOON SESSION DETAILS

Image of session:

image of session



Functions:

The function in C++ language is also known as procedure or subroutine in other programming languages.

To perform any task, we can create function. A function can be called many times. It provides modularity and code reusability.

Advantage of functions in C

There are many advantages of functions.

1) Code Reusability

By creating functions in C++, you can call it many times. So we don't need to write the same code again and again.

2) Code optimization

It makes the code optimized, we don't need to write much code.

Suppose, you have to check 3 numbers (531, 883 and 781) whether it is prime number or not. Without using function, you need to write the prime number logic 3 times. So, there is repetition of code.

But if you use functions, you need to write the logic only once and you can reuse it several times.

Types of Functions

There are two types of functions in C programming:

1. **Library Functions:** are the functions which are declared in the C++ header files such as `ceil(x)`, `cos(x)`, `exp(x)`, etc.
2. **User-defined functions:** are the functions which are created by the C++ programmer, so that he/she can use it many times. It reduces complexity of a big program and optimizes the code.

C++ Function Example

```
#include <iostream>
using namespace std;
void func() {
    static int i=0; //static variable
    int j=0; //local variable
    i++;
    j++;
    cout<<"i=" << i<<" and j=" <<j<<endl;
}
int main()
{
    func();
    func();
    func();
}
```

