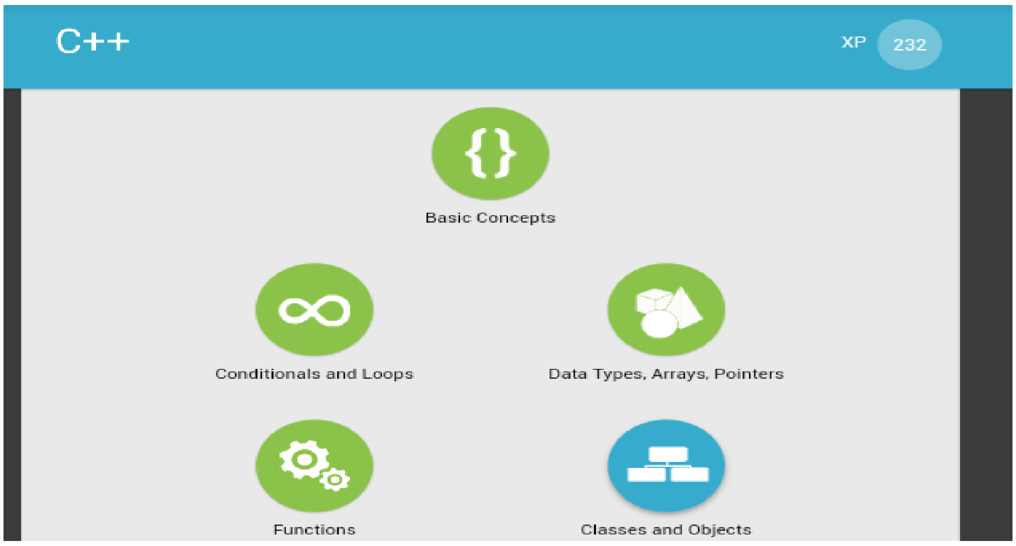
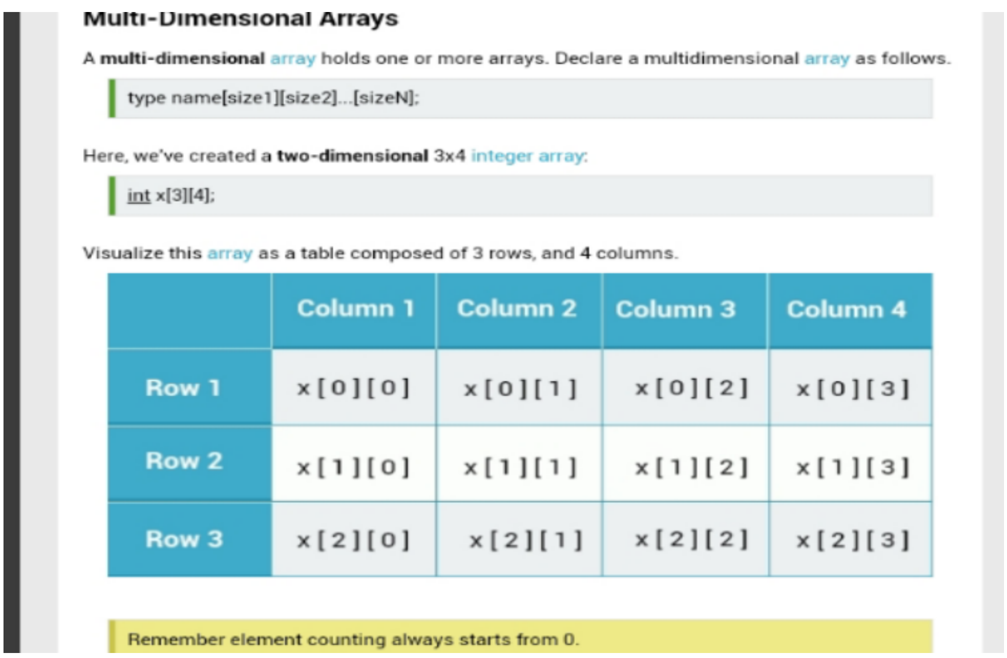
DAILY ASSESSMENT FORMAT

Date:	**23/06/2020**	**Name:**	Nichenametla Bhargavi
Course:	Programming in C++	**USN:**	**4AL17EC061**
Topic:	**1: Datatypes, Arrays and Pointer** **2: Functions**	**Semester & Section:**	**6th Sem A Sec**
Github Repository:	Bhargavi_Nichenametla		

SESSION

Image of the session





Multi-Dimensional Arrays

A **multi-dimensional** array holds one or more arrays. Declare a multidimensional array as follows.

```
type name[size1][size2]...[sizeN];
```

Here, we've created a **two-dimensional** 3x4 integer array:

```
int x[3][4];
```

Visualize this array as a table composed of 3 rows, and 4 columns.

	Column 1	Column 2	Column 3	Column 4
Row 1	x[0][0]	x[0][1]	x[0][2]	x[0][3]
Row 2	x[1][0]	x[1][1]	x[1][2]	x[1][3]
Row 3	x[2][0]	x[2][1]	x[2][2]	x[2][3]

Remember element counting always starts from 0.

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

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.

* For a function to use arguments, it must declare formal parameters, which are variables that accept the argument's values.

* 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>.

* 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.

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

* A recursive function in C++ is a function that calls itself.

* 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.

* There are two ways to pass arguments to a function as the function is being called.

1. 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.

2. 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.

Classes and Objects:

* Object Oriented Programming is a programming style that is intended to make thinking about programming closer to thinking about the real world. In programming, objects are independent units, and each has its own identity, just as objects in the real world do.

* In programming, an object is self-contained, with its own identity. It is separate from other objects. Each object has its own attributes, which describe its current state. Each exhibits its own behavior, which demonstrates what they can do.

* Objects are created using classes, which are actually the focal point of OOP.

* The class describes what the object will be, but is separate from the object itself.

* You can use the same class as a blueprint for creating multiple different objects. For example, in preparation to creating a new building, the architect creates a blueprint, which is used as a basis for actually building the structure.

* That same blueprint can be used to create multiple buildings.

* Programming works in the same fashion. We first define a class, which becomes the blueprint for creating objects. Each class has a name, and describes attributes and behavior.

* In programming, the term type is used to refer to a class name: We're creating an object of a particular type.

* Data abstraction is the concept of providing only essential information to the outside world.

* It's a process of representing essential features without including implementation details.

* Part of the meaning of the word encapsulation is the idea of "surrounding" an entity, not just to keep what's inside together, but also to protect it.
* In object orientation, encapsulation means more than simply combining attributes and behavior together within a class; it also means restricting access to the inner workings of that class.

* Class constructors are special member functions of a class. They are executed whenever new objects are created within that class.
* The constructor's name is identical to that of the class. It has no return type, not even void.

* Constructors can be very useful for setting initial values for certain member variables. A default constructor has no parameters.

* However, when needed, parameters can be added to a constructor.