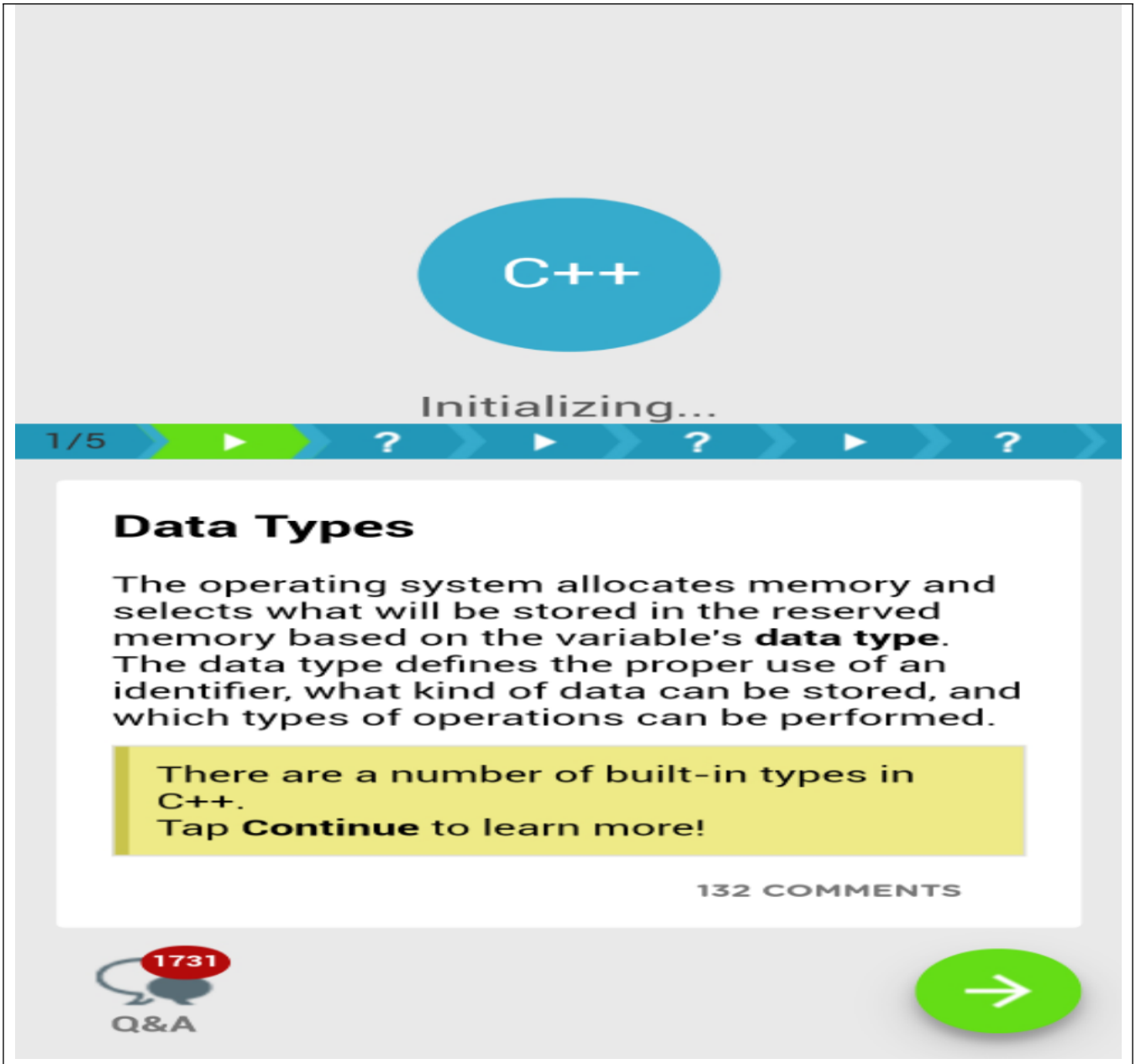
DAILY ASSESSMENT FORMAT

Date:	23-06-2020	Name:	Jagadeesha Hegde
Course:	C++ programming	USN:	4AL17EC036
Topic:	Data types,arrays,pointer	Semester & Section:	6th A-sec
Github Repository:	Jagadeesha-036		

FORENOON SESSION DETAILS
Image of session



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

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

o Code Optimization (less code)

o Random Access

o Easy to traverse data

o Easy to manipulate data

o Easy to sort data etc.

Disadvantages of C++ Array

o 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()

4. {

5. int arr[5]={10, 0, 20, 0, 30}; //creating and initializing array

6. //traversing array

7. for (int i = 0; i < 5; i++)

8. {

9. cout<<arr[i]<<"\n";

10. }

11. }
```

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.

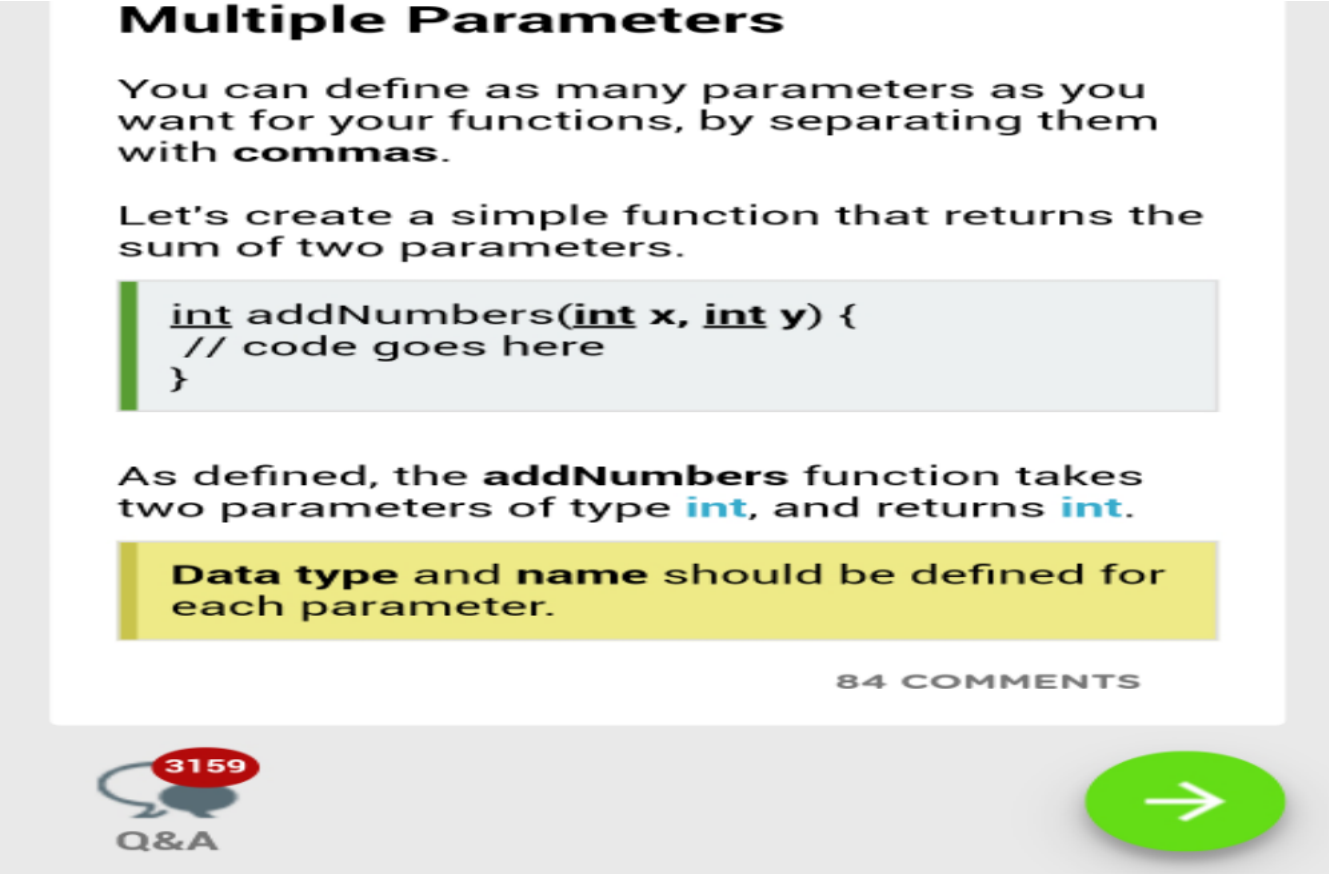
```
#include <iostream>
using namespace std;

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;
}
```

Date:	23-06-2020	Name:	Jagadeesha Hegde
Course:	C++ programming	USN:	4AL17EC036
Topic:	Functions	Semester & Section:	6th A-sec

AFTERNOON SESSION DETAILS

Image of session



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

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();
 }
```