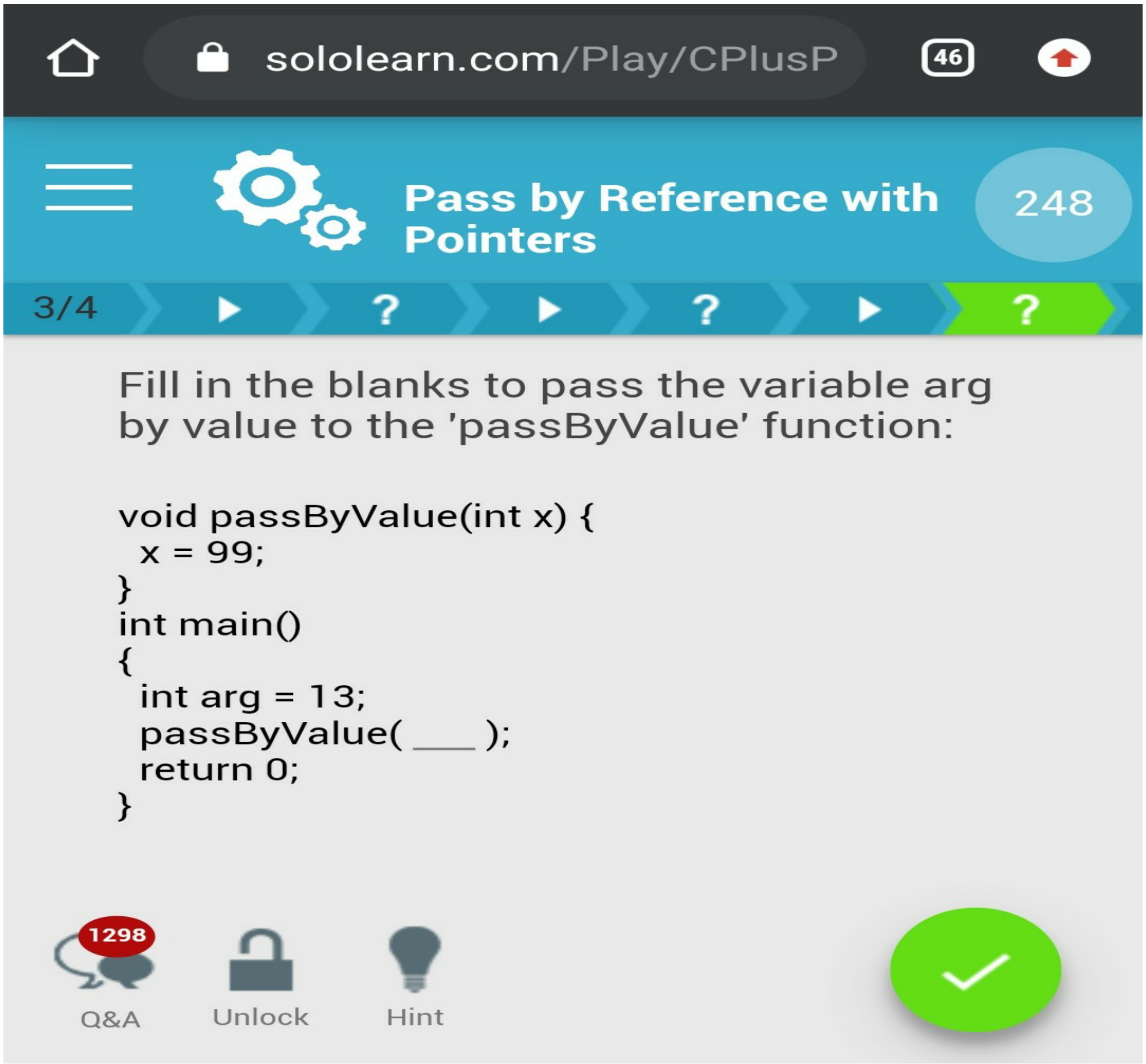
DAILY ASSESSMENT FORMAT

Date:	25-06-2020	Name:	Abhishek
Course:	C++ Programming	USN:	4al17ec001
Topic:	• Data Types , Arrays , Pointers • Functions	Semester & Section:	6 & 'A'
Github Repository:	Abhishek-online-courses		

SESSION DETAILS
Image of session



Fill in the blanks to pass the variable arg by value to the 'passByValue' function:

```
void passByValue(int x) {
  x = 99;
}
int main()
{
  int arg = 13;
  passByValue( ___ );
  return 0;
}
```

Report –

Module 3 : Data Types , Arrays , Pointers

- **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.
  - ✓ The basic data types are,

    1. int
    2. float
    3. double
    4. char
    5. string
    6. bool
- **Arrays**
  - ✓ An array is used to store a collection of data.
  - ✓ Instead of declaring multiple variables and storing individual values, we 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.
  - ✓ Example:

    ```
    int b[] = {11, 45, 62, 70, 88};
    ```
- **Pointer**
  - ✓ Pointer is a variable which holds the memory address of another variable.
  - ✓ 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.

Module 4 : Functions

- A **function** is a group of statements that perform a particular task.
- We can 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.

- **Define** a C++ function using the following syntax:

  ```
  return_type function_name( parameter list )
  {
    body of the function
  }
  ```

- **Function overloading** allows to create multiple functions with the same name, so long as they have different parameters.
- 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.
- There are two ways to pass arguments to a function as the function is being called. They are,
  - ✓ Pass by Value
  - ✓ Pass by Reference