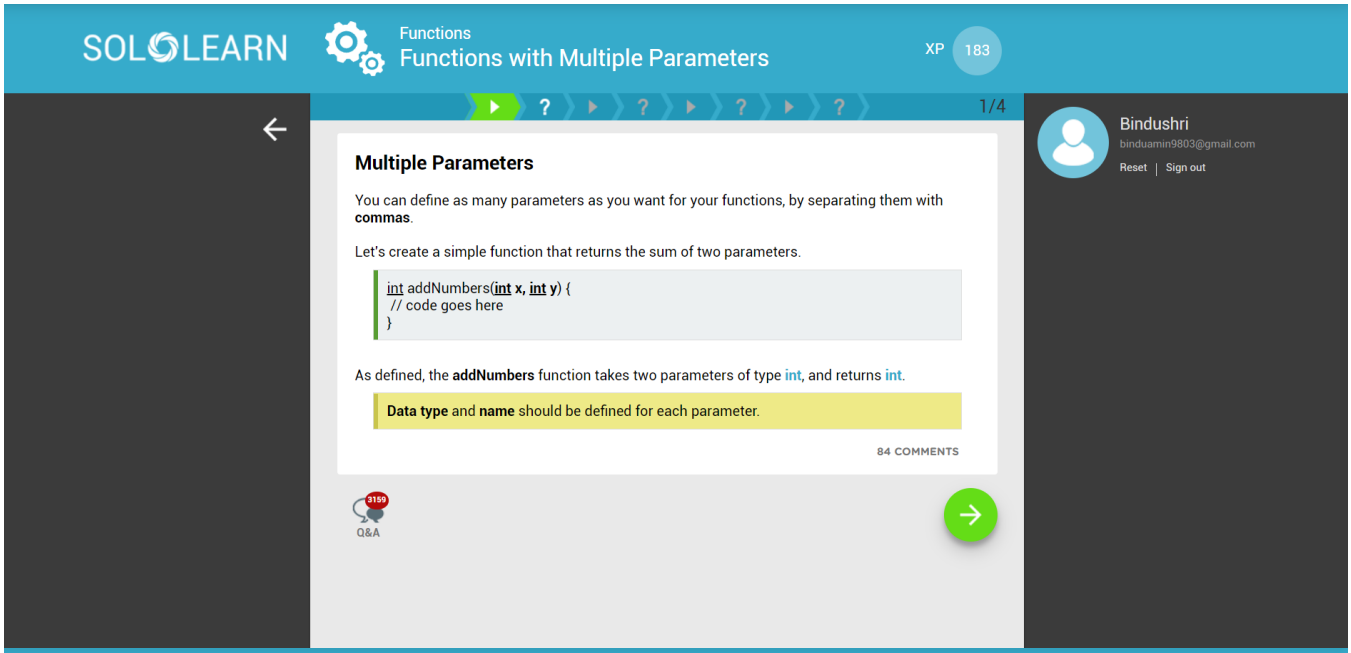
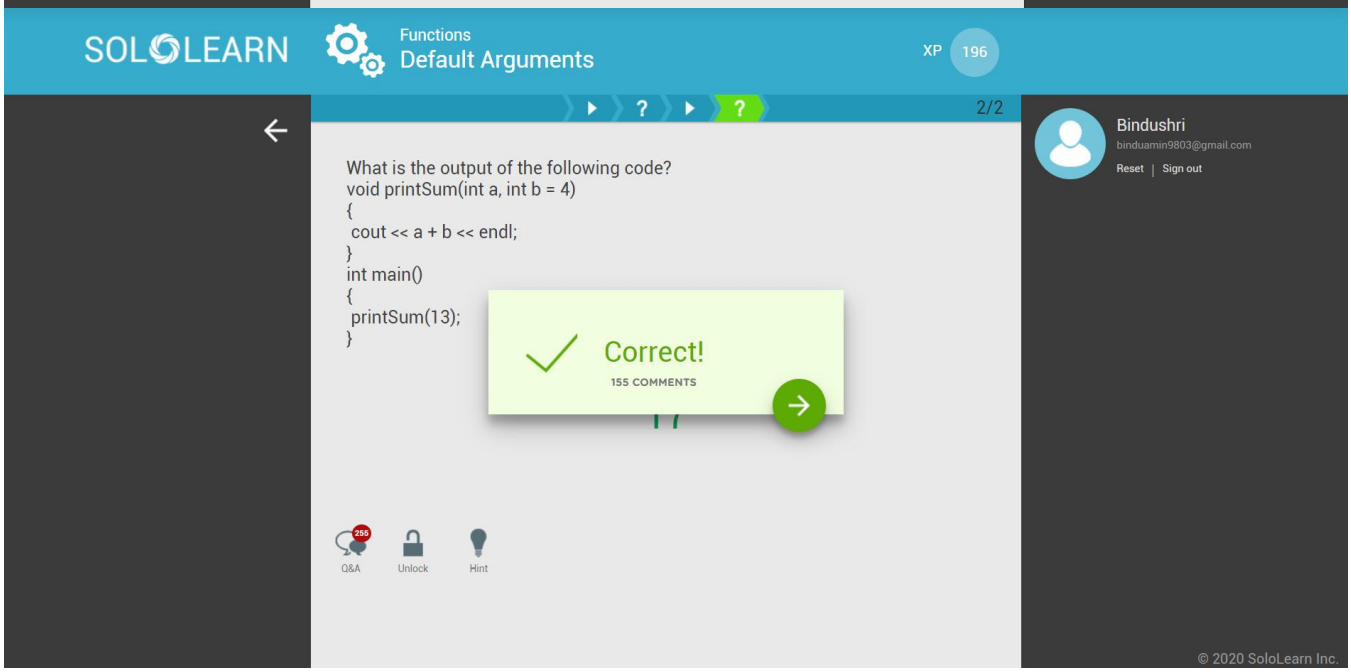
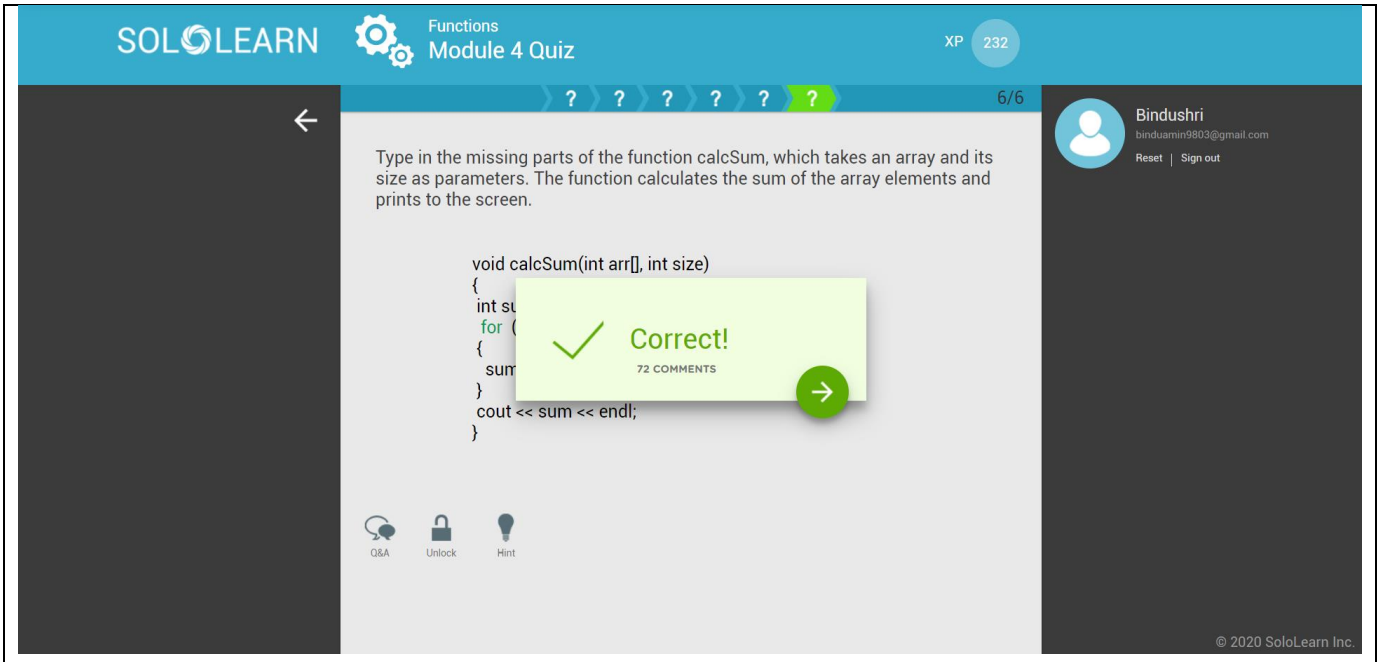
DAILY ASSESSMENT FORMAT

Date:	**23-06-2020**	**Name:**	**BINDUSHRI**
Course:	**C++ programming**	**USN:**	**4AL17EC011**
Topic:	**Datatypes,arrays,pointers functions**		**6th A**
Github Repository:	**Bindushri**		

FORENOON SESSION DETAILS






SOLOLEARN
Functions
Module 4 Quiz
XP 232
6/6
Bindushri
binduamin9803@gmail.com
Reset | Sign out
Type in the missing parts of the function calcSum, which takes an array and its size as parameters. The function calculates the sum of the array elements and prints to the screen.
void calcSum(int arr[], int size)
{
int su
for (
{
sum
}
cout << sum << endl;
}
Correct!
72 COMMENTS
Q&A
Unlock
Hint
© 2020 SoloLearn Inc.

...ers, Arrays, Pointers ✓

the operating system allocates memory and decides what will be stored in the reserved memory based on the variable's data-type.

Numeric Data type includes:
Integers (whole number), such as -7, 42.

floating point 3.14, -42.07

String & characters: composed of numbers, char or symbol. String literals are placed in double quotation marks. ex: "Hello", "My name is David" and similar

Characters are single letters or symbols & must be enclosed. b/co single quotes like 'a', 'b' etc.

Boolean data type returns just two possible value true() false()

int, float, double

Integer-type holds non-fractional number, which can be true or -ve.

```
int a=42;
```

:- can hold negative & +ve.
-> only the value of the default size the default size

```
#include<string>
using namespace std;
int main(){
string str= "I am learning C++";
return 0;
}
```

Variable naming Rules

Arrays

Array is used to store a collection of data, but it may be useful to think of an array as collection of variables that are all of the same type.

```
int a[5];
int b[5]={11,45,62,70,88};
```

using Arrays in loops.

```
int my_Arr[5];
for (int x=0; x<5; x++){
  My_Arr[x] = 42;
}
```

Arrays in calculation.

```
int arr[]= {11,35,62,555,
            98 a};
int sum=0;
for (int x=0; x<5; x++){
  sum+= arr[x];
}
cout<<sum<<endl
```

o/p 16652

Multi-Dimensional Array

A multi-dimensional array holds one or more arrays.

```
<type name[size][size]...[size];
```

Introduction to Pointers

Every variable in memory location which has its address defined

```
int score=5
cout<<&score<<endl
```

Dynamic - memory location cannot be performed without pointer.

```
int *ip;
double *dp;
float *fp;
char *ch;
```

Pointer operations:

Address-of operator (&): returns address of its operand.

Contents of (or dereference) operator (*):

Static & Dynamic Memory

Memory is divided into 2 types.

Stack: All local var take up memory from stack.

heap: unused program can be used while program runs dynamically allocate memory

delete memory -> deallocate new int -> allocate

Functions ✓

A function is a group of statements that perform a particular task

```
return_type function_name (para
                           ters)
{
  body of the function
}
```

```
#include<iostream>
using namespace std;
void print_something(){
  cout<<"Hi there!";
}
int main(){
  print_something();
  return 0;
}
```

also

```
#include<iostream>
using namespace std;
void print_something();
int main(){
  print_something();
  return 0;
}
void print_something(){
  cout<<"Hi there!";
}
```

Random numbers

Pseudo ~~num~~random number generator function that's called rand()

when used we are required to include <cstdlib>

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main(){
  cout<<rand();
}
```

srand() function is used to generate truly random numbers.

this allows to specify a seed value as its parameter which is used for rand()

```
int main(){
srand(98);
for (int x=1; x<=10; x++){
cout<<1+(rand()%6)<<
endl;
}
}
```

this makes use of time() function to get the number of seconds on your system then randomly seed the rand()

Default values

function over loading

Recursion

A recursive function is a function that calls itself.

$4! = 4 \times 3 \times 2 \times 1 = 24$

Pass by Reference with Pointers

Function arguments:-

there are two ways to pass arguments to a function as the function being called.

By value:

By reference:

X — X — X