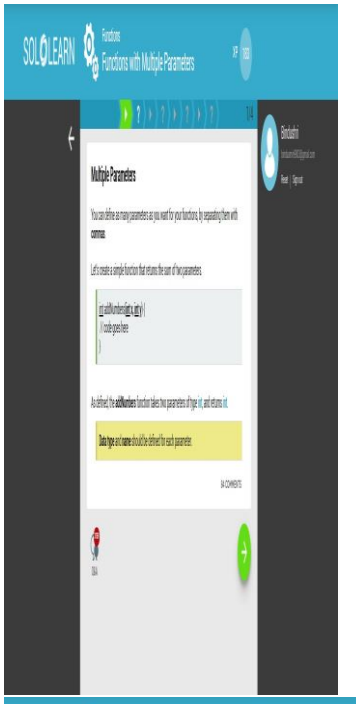
DAILY ASSESSMENT FORMAT

Date:	**23-06-2020**	**Name:**	porvi j gowda
Course:	**C++ programming**	**USN:**	AL17EC071
Topic:	**Datatypes,arrays,pointers functions**		h b
Github Repository:	porvi-2000		

FORENOON SESSION DETAILS


SOLOLEARN
Functions
Functions with Multiple Parameters
Multiple Parameters
You can define as many parameters as you want for your functions, by separating them with commas.
Let's create a simple function that returns the sum of two parameters
Data type and name should be defined for each parameter.

SOLOLEARN
Functions
Default Arguments
What is the output of the following code?
Correct!

SOLOLEARN
Functions
Module 4 Quiz
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.
Correct!

Types, Arrays, Pointers

the operating system allocates memory and selects 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, -42.67

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

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

Boolean data type returns just two possible value true(1) false(0)

Int, float, double

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

int a=42;
:- can hold negative & +ve.
→ only the values of the default size the default size

String char, ...

```
#include <string>
using namespace std;
int main(){
string[]= "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 myArr[5];
for (int x=0; x<5; x++){
  myArr[x] = 42;
}
```

Arrays in calculation.

```
int arr[]= {11, 35, 62, 555, 989};
int sum=0;
for (int x=0; x<5; x++){
  sum+= arr[x];
}
cout <<sum << endl
```

o/p 1652

Multi-Dimensional Array

A multi-dimentional array holds one or more arrays.

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

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_typefunctionname (parameter list)
{
  body of the function
}
```

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

also

```
#include <iostream>
using namespace std;
void printsomething();
int main(){
  printsomething();
  return 0;
}
void printsomething(){
  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 get to get the number of seconds on your system then it randomly seed the rand() function

Default values

function over loading

Recursion

A recursive function is a function that calls itself.

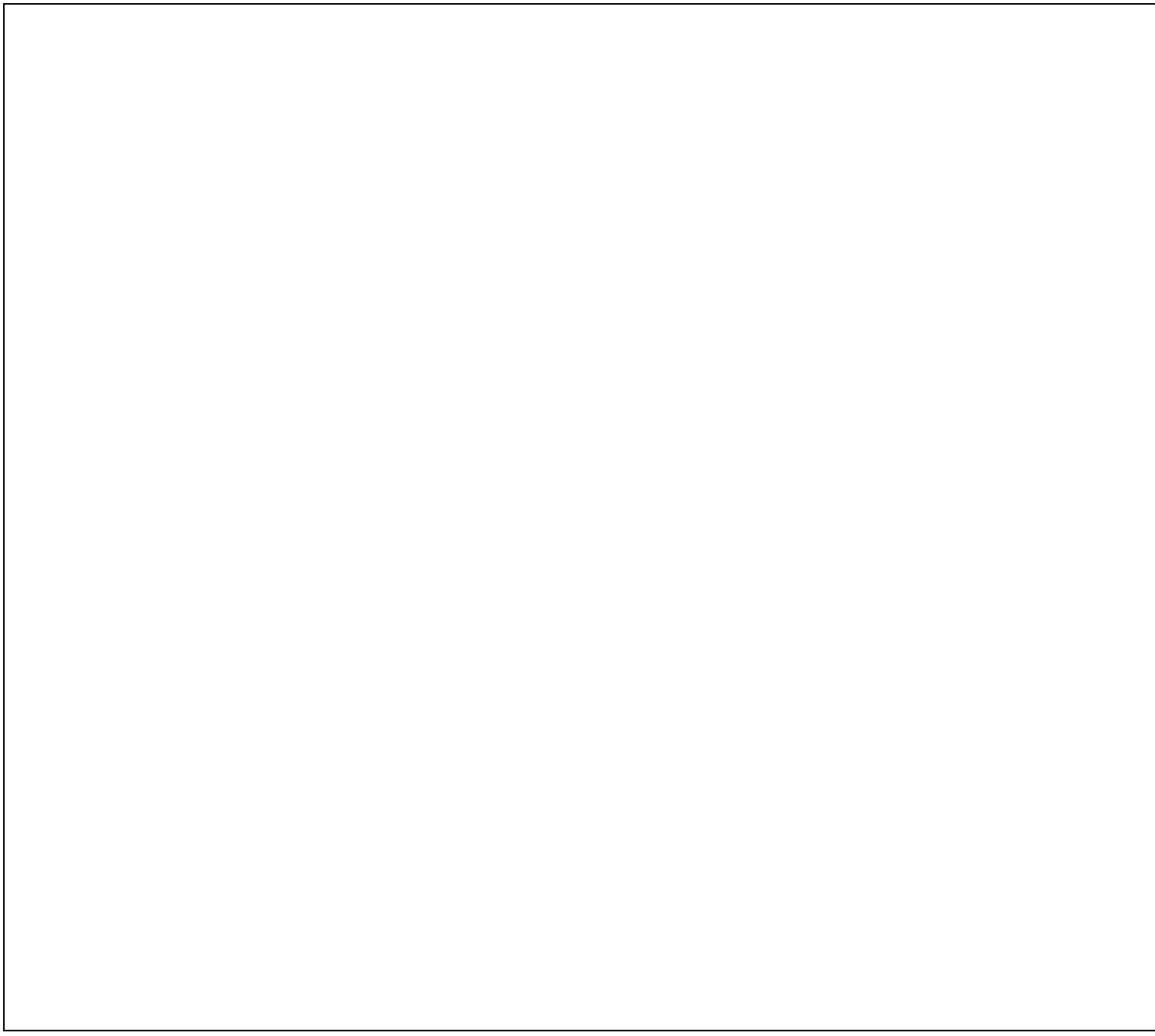
$4! = 4 \times 3 \times 2 \times 1 = 24$

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