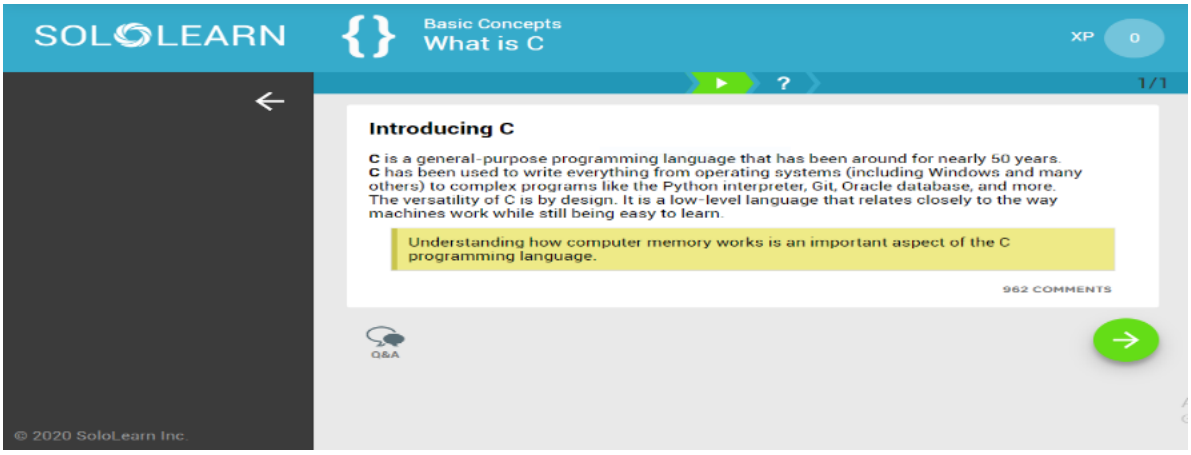
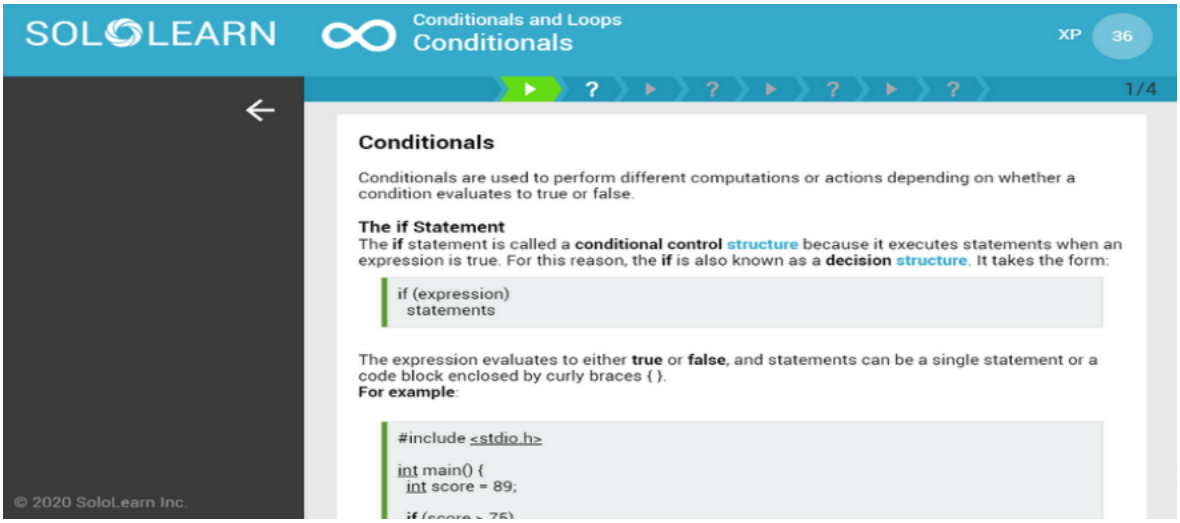
DAILY ASSESSMENT FORMAT

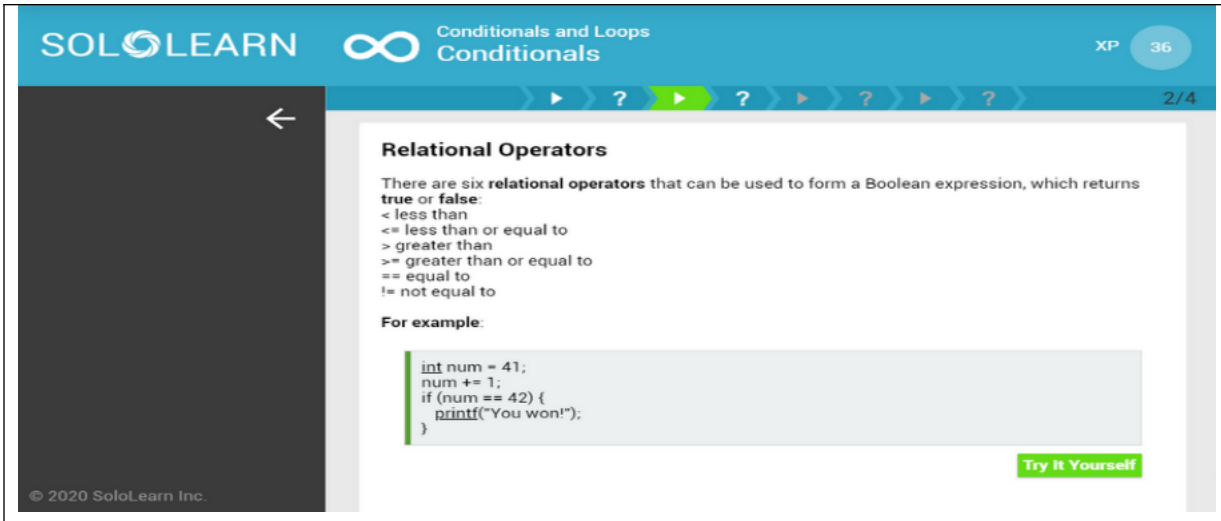
Date:	18-06-2020	Name:	Rajeshwari Gadagi
Course:	Solo Learn	USN:	4AL17EC076
Topic:	C Programming	Semester & Section:	6th SEM & 'B' SEC
Github Repository:	Rajeshwari-gadagi		

FORENOON SESSION DETAILS

Image of session







C Programming

Thursday - 18/06/2020

Modules:

→ Basic Concepts.

> What is C

Introducing C

C is a general purpose programming language that has been around for nearly 50 years. It has been used to write many things, from OS (including Windows & many others) to complex programs like python interpreter, Oracle database, etc.

> Hello world!

```
#include <stdio.h>
int main(){
  printf("Hello, world!\n");
  return 0;
}
```

printf function

The printf function is used to generate output.

```
#include <stdio.h>
int main(){
  printf("Hello, world!\n");
  return 0;
}
```

> Data types

C supports the following basic data types:

int: integer, a whole no.

float: floating point, a no. with a fractional part

double: double-precision floating point value.

char: single character

> Variables

A variable is a name for an area in memory. The name of a variable (also called identifier) must begin with either a letter or an underscore & can be composed of letter, digit & the underscore character.

> Constants

A constant is a value that cannot be changed from its initial assignment. By using constants with meaningful names, code is easier to read & understand. To distinguish constants from variables, a common practice is to use uppercase identifiers.

Another way to define a constant is with the #define preprocessor directive. The #define directive was mainly for defining constant values.

> Input

- C supports a number of ways for taking user input. getchar() Returns the value of the next single character input.
- The gets() function is used to read input as an ordered sequence of characters, also called a string. A string is stored in a char array.

> Output

Date: / /

- putchar() Output a single character.
- puts() function used to display output as a string.
  A string is stored in a char array.
- scanf() scans input that matches format specifier

> Formatted Input:
The scanf() function is used to assign input to variable. A call to this function reads input according to format specifier that convert input as necessary.
The scanf() statement waits for input & then makes assignment.

> Formatting Output:
The printf function was introduced in your very 1st "Hello World" program. A call to the function requires a format string which includes escape character sequences for outputting, special characters & format specifiers that are replaced by values.

> Comments
- Comments are explanatory information that you can include in a program to help for the reader of your code.
- Single-line comments: C++ introduced a double slash comment // as a way to comment single

> Arithmetic Operators:
C supports arithmetic operators + (addition), - (subtraction), * (multiplication), / (division), %. (modulus division).
Operators can often be used in assignment statements.

- Operator precedence:
The +, - are equal in precedence, as are *, /, & %. The *, /, %. are performed first in order from left to right & then +, -, also in order from left to right.

- Type conversion
When a numeric expression contains operands of different data types, they are automatically converted as necessary in a process called type conversion.

- Assignment Operator:
Evaluates the expression on the right side of the equal sign, first & then assigns that value to the variable on the left side of the =.

- Increment & Decrement:
Increment operator ++, adding 1 to a variable.
Decrement operator --, subtracting 1 from a vari

→ Conditionals & loops
> Conditionals are used to perform different computations or actions depending on whether a condition evaluates true or false.
- The if statement:
  if (expression)
    statement;

• Relational Operators:
<, less than, <= less than or equal to, > greater than, >= greater than or equal to, == equal to, != not equal to

• The if-else statement
The if statement can include an optional else clause that executes statements when an expression is false

• Conditional Expressions
Another way to form an if-else statement is by using the ?: operator in a conditional expression

> Nested if Statements:
An if statement can include another if statement to form a nested statement. Nesting an if allows a decision to be based on other requirements.

• The if-else if statement
When a decision among three or more actions is needed, the if-else if statement can be used. There can be multiple else if clauses & the last else clause is optional

> The switch statement
• The switch statement directs the program control by matching the result of an expression with a case value. There can be multiple values with unique labels

> Logical Operators:
Logical operators && and || are used to form a compound boolean expression that tests multiple conditions. A third logical operator is ! used to reverse the state of a boolean expression.

The && operator.
The logical AND operator && returns true result only when both of its operands are true

> The || operator.
The logical OR operator || returns a true result when any one of its operands or both of its operands are true

The ! Operator
The logical NOT operator ! returns the reverse of its value. NOT true returns false, & NOT false returns true.

The while loop
while (expression) { statements }

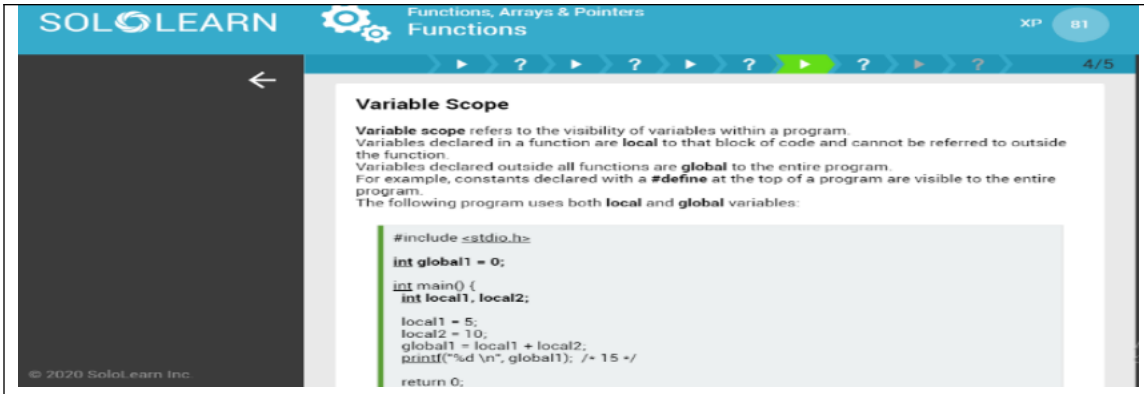
> The do-while loop
do { statements.
while (expression);
}

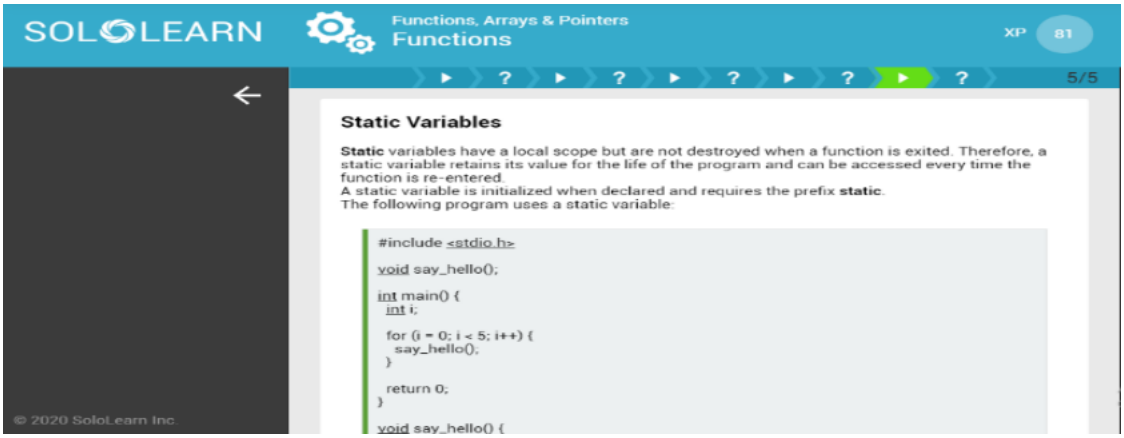
> break & continue
• The break statement was introduced for use in switch statement. It is also useful for immediately exiting a loop.
• The continue when you want to remain in the loop, but skip ahead to the next iteration, you use the continue statement

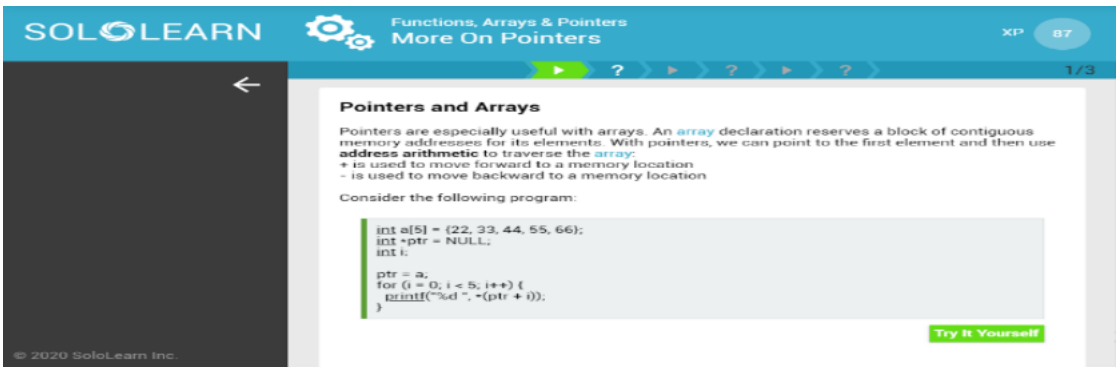
The for-loop
for (init value; condition; increment) { statements }

Date:	18-06-2020	Name:	Rajeshwari Gadagi
Course:	Solo Learning	USN:	4AL17EC076
Topic:	C Programming	Semester & Section:	6th SEM & 'B' SEC
Github Repository:	Rajeshwari-gadagi		

AFTERNOON SESSION DETAILS
Image of session


SOLOLEARN
Functions, Arrays & Pointers
Functions
XP 81
4/5
Variable Scope
Variable scope refers to the visibility of variables within a program.
Variables declared in a function are local to that block of code and cannot be referred to outside the function.
Variables declared outside all functions are global to the entire program.
For example, constants declared with a #define at the top of a program are visible to the entire program.
The following program uses both local and global variables:
#include <stdio.h>
int global1 = 0;
int main() {
int local1, local2;
local1 = 5;
local2 = 10;
global1 = local1 + local2;
printf("%d \n", global1); /* 15 */
return 0;
© 2020 SoloLearn Inc


SOLOLEARN
Functions, Arrays & Pointers
Functions
XP 81
5/5
Static Variables
Static variables have a local scope but are not destroyed when a function is exited. Therefore, a static variable retains its value for the life of the program and can be accessed every time the function is re-entered.
A static variable is initialized when declared and requires the prefix static.
The following program uses a static variable:
#include <stdio.h>
void say_hello();
int main() {
int i;
for (i = 0; i < 5; i++) {
say_hello();
}
return 0;
}
void say_hello() {
© 2020 SoloLearn Inc


SOLOLEARN
Functions, Arrays & Pointers
More On Pointers
XP 87
1/3
Pointers and Arrays
Pointers are especially useful with arrays. An array declaration reserves a block of contiguous memory addresses for its elements. With pointers, we can point to the first element and then use address arithmetic to traverse the array:
+ is used to move forward to a memory location
- is used to move backward to a memory location
Consider the following program:
int a[5] = {22, 33, 44, 55, 66};
int *ptr = NULL;
int i;
ptr = a;
for (i = 0; i < 5; i++) {
printf("%d ", *(ptr + i));
}
Try It Yourself
© 2020 SoloLearn Inc

C Programming

Thursday - 18/06/2020

> Functions in C -

Function are critical to C Programming & are used to accomplish a program whether it is a small or large one.
main() function & printf() function.
A function :
- is a block of code that performs a specific task
- is reusable
- makes a program easier to test.
- can be modified without changing the calling program.

return type function name (parameters);

> Function Prototype
When the parameter type and names are included in declaration, the declaration is called a function prototype

> Function Parameters
A function's parameters are used to receive values required by the function. Values are passed to these parameters as arguments through function call

> Variable Scope
It refers to the visibility of variables within a program.
Variables declared in a function are local & outside a function are global.
It differ.

> Static Variables
Static variables have a local scope but are not destroyed when a function is exited.
A static variable is initialized when declared & requires the first time.

> Recursive Functions
A recursive function is one that calls itself & includes some code, an exit condition & ending the recursive calls. In the case of finding a factorial, the base case is num equal to 1.

> Array in C
- An array is a data structure that stores a collection of related values that can all be sometimes
- Accessing Array Elements
The contents of an 'array' are called elements. Each element can be identified by an index no.
- using loops with arrays

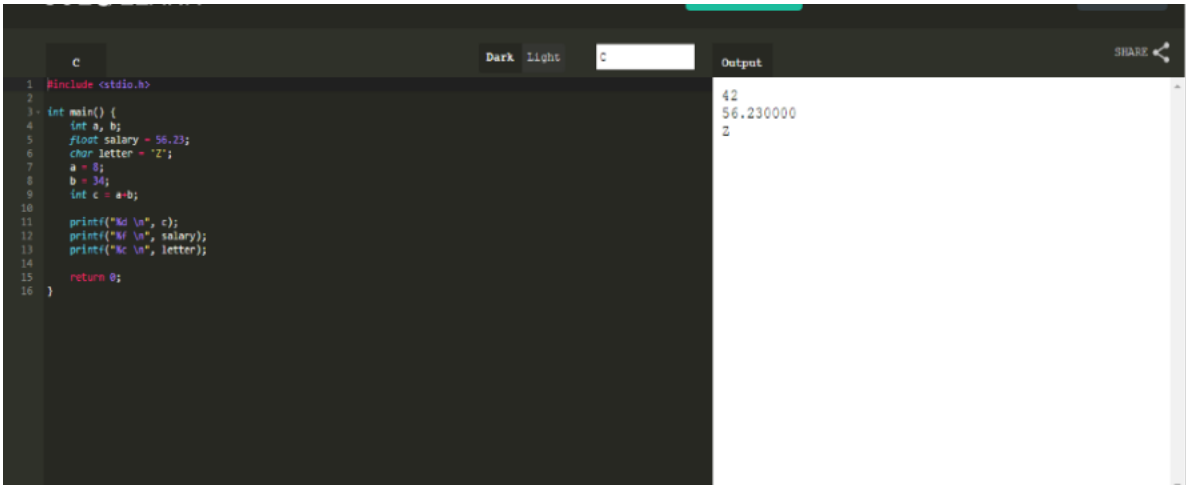
> Two-dimensional array
A 2-dimensional array is an array of arrays & can be thought of as a table.

> Pointers
- using Memory
- what is a pointer?
Pointers are very important in C programming because they allow you to directly work with memory locations.
They are fundamental to arrays, strings & other data structures & algorithms.
pointer_type * identifier;

> Pointers in Expressions
Pointers can be used in expressions just as any variable. Arithmetic operations can be applied to whatever the pointer is pointing to.

> More on Pointers
- Pointers are especially useful with Arrays.
It is useful to move forward to an array location
- is used to move backward to an array location
- More Address Arithmetic
Address arithmetic can also be thought of as pointer arithmetic because the operations involve pointers.
- Pointers & Functions

> Functions & Arrays
An array cannot be passed by value to a function. However, an array name is a pointer, so just passing an array name to a function is passing a pointer to the array.
> Functions that Return an Array


C
Dark
Light
Output
SHARE
#include <stdio.h>
int main() {
int a, b;
float salary = 56.23;
char letter = 'Z';
a = 8;
b = 34;
int c = a+b;
printf("%d \n", c);
printf("%f \n", salary);
printf("%c \n", letter);
return 0;
}
42
56.230000
Z