

DAILY ASSESSMENT

| | | | |
|--------------------|--|---------------------|---------------------|
| Date: | 18 June 2020 | Name: | Sushmitha R Naik |
| Course: | C Programming | USN: | 4AL17EC090 |
| Topic: | <ul style="list-style-type: none">• Basic Concept• Conditionals & Loops• Functions, Array & Pointers• Strings & Function Pointers | Semester & Section: | 6 th & B |
| GitHub Repository: | Sushmitha_naik | | |

The image displays two screenshots of the Sololearn Code Playground interface, showing C code being executed.

Top Screenshot:

- Language: C
- Code:

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Hello, World!\n");
5     return 0;
6 }
```
- Output: Hello, World!
- Buttons: SAVE, RESET, SAVE AS, RUN
- Download links: Google play, App Store

Bottom Screenshot:

- Language: C
- Code:

```
1 #include <stdio.h>
2
3 int main() {
4     int a, b;
5     float salary = 56.23;
6     char letter = 'Z';
7     a = 8;
8     b = 24;
9     int c = a+b;
10
11     printf("%d \n", c);
12     printf("%f \n", salary);
13     printf("%c \n", letter);
14
15     return 0;
16 }
```
- Output:

```
42
56.230000
Z
```
- Buttons: SAVE, RESET, SAVE AS, RUN
- Download links: Google play, App Store

Report:

C Programming:

- C programming basic commands to write a C program
- A simple C program with output and explanation
- Steps to write C programs and get the output
- Creation, Compilation and Execution of a C program
- How to install C compiler and IDE tool to run C programming codes
- Basic structure of a C program
- Example C program to compare all the sections
- Description for each section of the C program
- C programs ([Click here for more C programs](#)) with definition and output – C program for Prime number, Factorial, Fibonacci series, Palindrome, Swapping 2 numbers with and without temp variable, sample calculator program and sample bank application program etc.
- If you want to create, compile and execute C programs by your own, you have to install C compiler in your machine. Then, you can start to execute your own C programs in your machine.
- You can refer below link for how to install C compiler and compile and execute C programs in your machine.
- Once C compiler is installed in your machine, you can create, compile and execute C programs as shown in below link.
- If you don't want to install C/C++ compilers in your machine, you can refer online compilers which will compile and execute C/C++ and many other programming languages online and display outputs on the screen. Please search for online C/C++ compilers in Google for more details.

Conditionals & Loops:

1. if statement

- This is the most simple form of the branching statements.
- It takes an expression in parenthesis and an statement or block of statements. if the expression is true then the statement or block of statements gets executed otherwise these statements are skipped.

2. switch statement:

- The switch statement is much like a nested if .. else statement. Its mostly a matter of preference which you use, switch statement can be slightly more efficient and easier to read.

3. Using break keyword:

- If a condition is met in switch case then execution continues on into the next case clause also if it is not explicitly specified that the execution should exit the switch statement. This is achieved by using break keyword.

4. while loop

- The most basic loop in C is the while loop. A while statement is like a repeating if statement. Like an If statement, if the test condition is true: the statements get executed. The difference is that after the statements have been executed, the test condition is checked again. If it is still true the statements get executed again. This cycle repeats until the test condition evaluates to false.

5. for loop

- for loop is similar to while, it's just written differently. for statements are often used to process lists such a range of numbers:

6. do...while loop

- do ... while is just like a while loop except that the test condition is checked at the end of the loop rather than the start. This has the effect that the content of the loop are always executed at least once.

7. break and continue statements

- C provides two commands to control how we loop:
- break -- exit from loop or switch.
- continue -- skip 1 iteration of loop.

Functions, Array & Pointers:

- A function is a group of statements that together perform a task. Every C program has at least one function, which is main(), and all the most trivial programs can define additional functions.
- You can divide up your code into separate functions. How you divide up your code among different functions is up to you, but logically the division is such that each function performs a specific task.
- A function declaration tells the compiler about a function's name, return type, and parameters. A function definition provides the actual body of the function.
- The C standard library provides numerous built-in functions that your program can call. For example, strcat() to concatenate two strings, memcpy() to copy one memory location to another location, and many more functions.
- Arrays a kind of data structure that can store a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.
- Instead of declaring individual variables, such as number0, number1, ..., and number99, you declare one array variable such as numbers and use numbers[0], numbers[1], and ..., numbers[99] to represent individual variables. A specific element in an array is accessed by an index.
- All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.
- A pointer is a variable whose value is the address of another variable, i.e., direct address of the memory location. Like any variable or constant, you must declare a pointer before using it to store any variable address.

Strings & Function Pointers:

- **Strings are actually one-dimensional array of characters terminated by a null character '\0'. Thus a null-terminated string contains the characters that comprise the string followed by a null.**
- **The following declaration and initialization create a string consisting of the word "Hello". To hold the null character at the end of the array, the size of the character array containing the string is one more than the number of characters in the word "Hello."**
- **The first question that may come to your mind is why would we use pointers to call a function when we can simply call a function by its name: function(); - that's a great question! Now imagine the sort function where you need to sort an array. Sometimes you want to order array elements in an ascending order or descending order. How would you choose? Function pointers!**