

GUIDED INTRODUCTION TO PROGRAMMING

Introduction and the Print and Scan Functions

Overview:

This guide was created to help students get an early introduction to programming to create a spark so that students may be interested in pursuing a career in STEM and to help with programming in the future. This lesson will introduce students to programming and some terminology that goes along with programming.

Learning Outcomes

- Students will explain the basic concepts programming and coding
- Students will perform basic programming and coding theories
- Students will apply programming and coding procedures to hands on experience exercise
- Students will formulate simple code to get the console to display text and read input.
- Students will utilize the print and scan functions to prompt the user to input values and make a response to the user with the print function.

Materials

- Computer with internet connection
- Website for the online compiler <https://replit.com/languages/c>
- Lesson 1 Lesson Plan
- Vocabulary Sheet

The Activity

Students will be introduced to the concept of programming using an online compiler to observe and then create simple programs using code.

Part 1

Let's start with an answer to a question that some of you may have, "What is programming?" **Programming** is the process of creating instructions that tell a computer how to perform a task. We do this using what we call code. **Code** is language we use to create the instructions while programming.

There are hundreds of modern code languages that we use to do different things on computers. You will be focusing on one programming language.

First, start with a very simple introduction to a programming language called C. Question- “Why is it called ‘C’ that’s a boring name?” While the name is a bit basic it is one of the fundamental programming languages that most, if not all, programmers need to learn. This is because all modern code can trace some of its roots back to C.

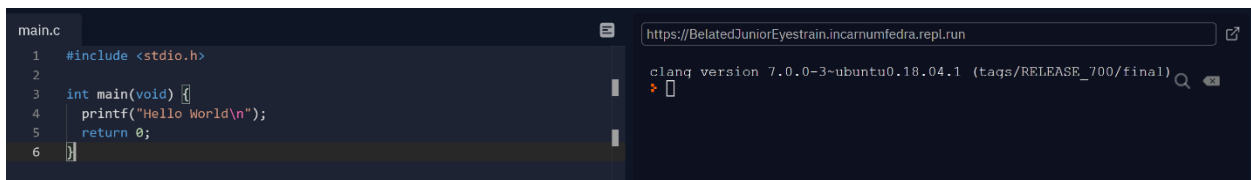
Print and Scan

This lesson will be on the print and scan function in C. These two functions are the first you usually use when beginning to program. The **print function** will take what you put into the function and will **output**, or print, it to the computer screen. Along with this the **scan function** will take an **input** from the user and will save the input in **variables**. By the end of the lesson you will learn to utilize the print and scan functions to prompt the user to input some values and make a response to the user with the print function.

In order to get hands on with programming in C we will be using an online **compiler**, please click or copy this link to get to a free online compiler <https://repl.it/languages/c>. This will be used to practice our code. While this looks very similar to what we use to program with, it is made to work online and can vary in some areas, but this gives you a good idea of the tools a programmer uses to do their job.

Part 2

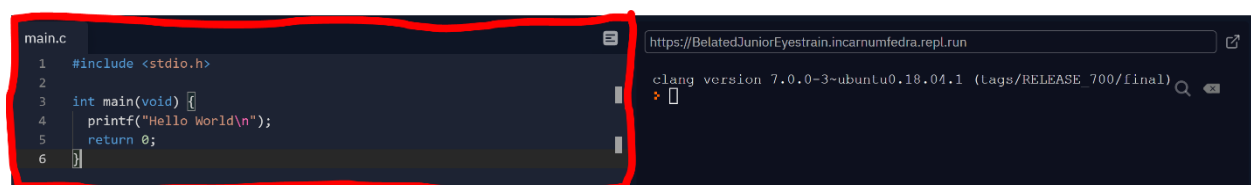
Now this is what you should see when you start up the compiler.



```
main.c
1 #include <stdio.h>
2
3 int main(void) {
4     printf("Hello World\n");
5     return 0;
6 }
```

clang version 7.0.0-3-ubuntu0.18.04.1 (tags/RELEASE_700/final)

There is already some code there for you to look at. However, before we start, we need to know what we are looking at.

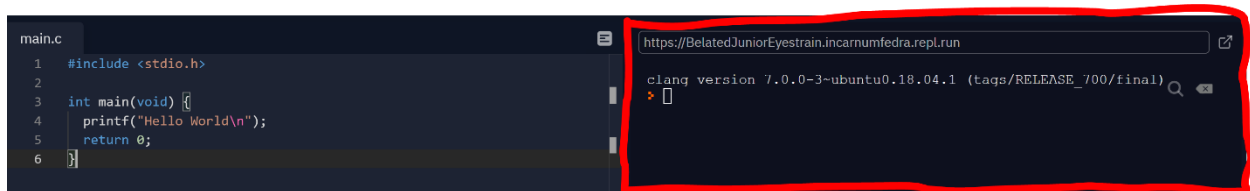


```
main.c
1 #include <stdio.h>
2
3 int main(void) {
4     printf("Hello World\n");
5     return 0;
6 }
```

clang version 7.0.0-3-ubuntu0.18.04.1 (tags/RELEASE_700/final)



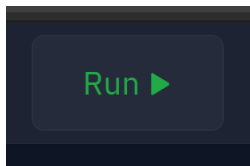
This left side is where we will be inputting our code. It is very important to not delete what is in the code right now. If you happen to delete something important, don't worry just take a look back at the code in the pictures and just copy it in, or we can just restart by refreshing the page and that will bring you back to the original code. Doing that will put you back where we started program.



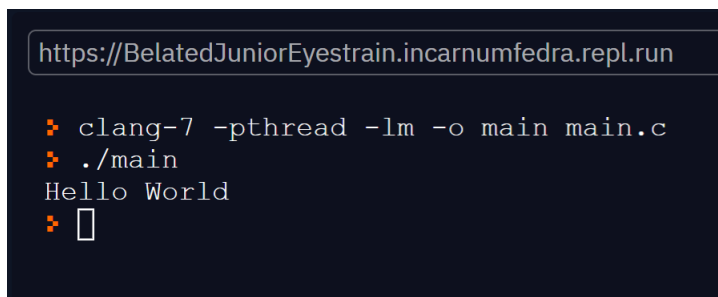
```
main.c
1 #include <stdio.h>
2
3 int main(void) {
4     printf("Hello World\n");
5     return 0;
6 }
```

```
https://BelatedJuniorEyestrain.incarnumfedra.repl.run
clang version 7.0.0-3-ubuntu0.18.04.1 (tags/RELEASE_700/final)
> □
```

This right side is where we will see the output of our code. This is where we will see the output of our code when we run it. What the printf function will do is print whatever text you put in it to the console. To test it on the program, hit the big "Run" button at the top.



After you run the program you should get this output.



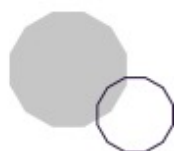
```
https://BelatedJuniorEyestrain.incarnumfedra.repl.run
❖ clang-7 -pthread -lm -o main main.c
❖ ./main
Hello World
❖ □
```

And with that Congratulations, you have just run your first working program.

The program you just ran is what most programmers saw when they started. It is known as the "Hello World" program, the universal start point for most programmers. It's an appropriate name as all it does is it outputs the statement "Hello World".

This is done using the printf function which is formatted like so:

```
printf("Blah blah blah\n");
```



The `\n` is used in the `printf` function to start a new line, we mainly use this to keep our output nice and neat, otherwise it is just a jumbled mess.

Part 3

Now you can start to make your own code.

There are a few things to take into consideration when writing code. First, and most importantly, after you finish writing a line of code you need to end that line with a **semicolon** which looks like this `;` on your keyboard.

For example, we are going to do a bit of code that will allow you to type your name into the **console** and the computer will say hello. Some of the code will be taught later on so to start here is some of the full code that you will use.

```
int namechar = 0;
char name[namechar];
printf("Hello World\n");

printf("How many letters are in your name?\n");

//This is where you scan in how many letters are in your name

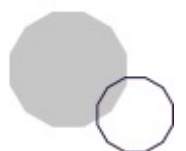
printf("Now that we know how many letters are in your name, please type in your first name.\n");

//This is where you scan in your name

printf("Hello %s it is nice to meet you.\n", name);
return 0;
```

Remember to be sure to put this in between the `{` and the `}` of the `int main` function for the code to work. These braces are extremely important to code as they are used to contain instructions for functions and programs we make. Think of them like writing on a test, if you write everything on the test paper your teacher will be able to grade it, but if some of it goes on your desk, your teacher cannot grade what you wrote off the test and on your desk.

Next, you will use the `scanf` function. The `scanf` function allows users to input values for variables and will save them to the appropriate variable. Now, the `scanf` function is a bit more in depth than the `printf` function is because now we have to use the variable we create.



First, one must understand variables. There is an array of variables that each have their own traits and uses. Today the only variables you need to know for now is the “int” and “char” variables that we will be using for our scanf. An “int” is an integer, you may or may not have learned about integers in your math class but if you haven’t, an integer is just a simple number that can be either positive or negative. A “char” is a variable that holds a character like from the alphabet or punctuation.

Now that we know about the variables we will be using I can now show you how a scanf function is formatted:

```
scanf("%variablecaller", &variablename);
```

This is the exact formatting you use to make a scanf function work.

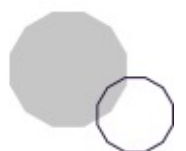
For today’s lesson though we will need two scanf functions. One for an int and one for a char

The variable caller for an int is “%d” and the variable caller for the char in this case is “%s”.

Now you try and input the scanf into your code and get it to function. You may need to try a few things before you get it to run properly but don’t get discouraged. If you need help ask your teacher or look up how code works on the internet, there are hundreds of great sites that can help you with coding. Sites that are very helpful are places like Stack Overflow, Geeksforgeeks, and W3 Schools. All of these sites have great explanations on coding principles along with good visuals to help you understand how some things work. Stack Overflow is especially helpful as it is a free online forum where people can ask for help with code, the people who use it are usually really nice and help you fix your code.

The way you know your code was done correctly is when your console looks like this after you run and enter what it asks for.

```
❖ clang-7 -pthread -lm -o main main.c
❖ ./main
Hello World
How many letters are in your name?
4
Now that we know how many letters are in your name, please type in your first name.
Ryan
Hello Ryan it is nice to meet you.
❖ █
```



And with that you have completed your first lesson and introduction into the world of coding.

Assessment

1. Attempt the code above to scan in your name and have the computer say hello.
2. Go to [w3schools.com](https://www.w3schools.com) and write down 3 languages that they have tutorials for.
3. What are 3 other programming languages other than the 3 you saw on [w3schools.com](https://www.w3schools.com)

Additional Resources

- <https://www.w3schools.com/>
- <https://www.geeksforgeeks.org/c-programming-language/>
- <https://replit.com/languages/c>

Created By:
Ryan Floyd, Computer Science, '22

© Florida Polytechnic University, 2020. No part of the materials available may be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine-readable form, in whole or in part, without prior written consent of Florida Polytechnic University. Any other reproduction in any form without the permission of Florida Polytechnic University is prohibited.

Thank you for downloading this lesson, please take a moment to complete our [survey](#)

