

# FLORIDA POLYTECHNIC

# INTRODUCTION TO PROGRAMMING

Teachers Guide

## Introduction to Programming

### Lesson 1

### Teachers Edition

### Print and Scan

Hello and welcome to a guided introduction to programming. I created this guide to help get students an early introduction to programming as soon as possible to get them interested in pursuing a career in STEM and to help them in the future. This guide will mirror the student guide but will have added notes on what to look out for and how to help them along if they encounter errors.

First, 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. Today we will be focusing on one programming language.

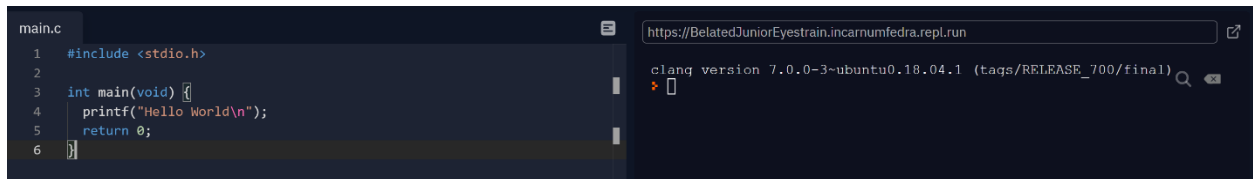
So today let us start with a very simple introduction to a programming language called C. Now you may be wondering, “Why is it called ‘C’ that’s a boring name?” and while the name is a bit basic it is one of the fundamental programming languages that most, if not all, programmers need to learn. This is because all modern code can trace some of its roots back to C.

Today’s 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.

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>. We will be using this 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.

This is what you should see when you start up the compiler.



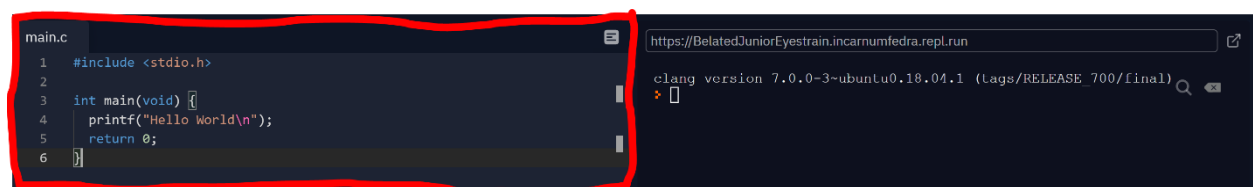
Be sure that the students **do not delete or add** anything when they get to this page.

If they add anything that is not on this page just tell them to delete it

If they delete anything tell them just to copy over exactly what is in the picture above

If students are still having issues. Have them close the compiler and open the link again.

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



This is where you will be helping students the most, their errors will (for the most part) stem from their code.

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, do not worry, just look back at the code in the pictures and just copy it in, or we can just restart by hitting the “+” in the top right corner and choosing the C option and hit the “Create repl” button. 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 }

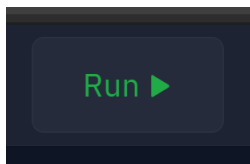
https://BelatedJuniorEyestrain.incarnumfedra.repl.run
clang version 7.0.0-3-ubuntu0.18.04.1 (tags/RELEASE_700/final)
>
```

Errors will appear when the code is run and the compiler hits an error. The console on the right is a good way of trying to determine what is wrong with the code but can become confusing for newcomers to programming. There are three types of errors you will need to be on the lookout for to help the students:

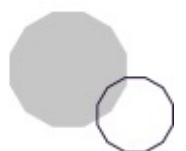
- 1.) Syntax Errors: Pictured above ^ Include: misspelling functions, missing punctuation, etc...
- 2.) Runtime Errors: Errors that occur during running the code Include: not calling or improperly calling functions, missing variable callers in scan or print
- 3.) Logic Errors: This should not occur in this lesson but it may happen in future lessons, Errors that provide incorrect outputs based on flawed logic in code.

Errors in the code will be underlined in either a green or red line, hovering over the code can help to give an idea of what went wrong and how to fix it.

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
❖ □
```

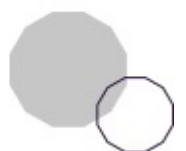
The code to get this output is the exact same as what was pictured previously.

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 starting point for most programmers. It is an appropriate name as all it does is it outputs the statement “Hello World”. Now that we have that introduction to what code looks like and what the output looks like we can now start to make our 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. 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.

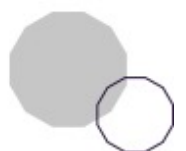
Now we will learn about the scanf function. The scanf function allows users to input values for variables and will save them to the appropriate variable. Now, the scanf function is a bit more in depth than the printf function is because now we must use the variable we create.

To start, focus on variables. There are an array of variables, and 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 have not, 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.

Using what you know about the variables, we can cover how a scanf function if formatted:

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

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



Today's lesson 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".

Next, you will 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 do not get discouraged. 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. 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 there to 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.

If the students code is greyed out and this line is still there:

```
//This is where you scan in your name
```

Just delete that line of code, it is a comment and can throw off the code if anything is put in the same line.

This is what the code should look like:

```
main.c
1  #include <stdio.h>
2
3  int main(void) {
4      int namechar = 0;
5      char name[namechar];
6      printf("Hello World\n");
7
8      printf("How many letters are in your name?\n");
9
10     scanf("%d", &namechar);
11
12     printf("Now that we know how many letters are in your name, please
13     type in your first name.\n");
14     scanf("%s", name);|
15
16     printf("Hello %s it is nice to meet you.\n", name);
17     return 0;
18
19 }
```



A few key things to check if the student is running into errors.

- 1.) Be sure they have semicolons in the right place
- 2.) The second scanf is a bit different than the first and does not have a '&' in it
- 3.) All code is contained in the int main(void){ all code }
- 4.) Compare students code to the code pictured above for any other issues and try to guide them towards this answer.

If a student's code does not look like what is above but works properly, there is nothing to worry about, answers to a programming problem can come in many forms.

```
❖ 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 yo
ur 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.  
Congratulations!

Created By:  
Ryan Floyd, Computer Science, '22

© Florida Polytechnic University, 2021. 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](#)

