**QUIZ**

Why won't the function call work?

```
#include <iostream>

void translate_to_human(std::string cat_sounds) {

  std::cout << cat_sounds << " roughly translates to 'I want food. Give me food.'";

}

int main() {

  translate_to_human();

}
```

The function has the wrong return type; it should be `std::string`.

The function needs a return value.

The function has a parameter, so the function call needs an argument.

"Meow" would work as an argument. It would roughly translate to "I want food. Give me food."

What part of the program has access to the value of `good`?

```
#include <iostream>

void good_or_not(bool good) {

  if (good) {

    std::cout << "Good dog!";

  } else {

    std::cout << "Let's work on that...";

  }

}

int main() {

  good_or_not(true);

}
```

Throughout the program.

Only inside `good_or_not()`.

👏 `good` is defined within the function's local scope so it cannot be accessed outside of the function.

How many parameters does this function have?

```
int add_fruit(int pineapples, int oranges, int bananas) {

  return pineapples + oranges + bananas;

}
```

3

👏 The parameters are `pineapples`, `oranges`, and `bananas`.

When would it make sense to build a function template?

You want to sometimes use three parameters and sometimes two parameters.

You want to use the same function body but with different data types.

Function templates allow programmers to add data types as parameters.

One of your parameters is usually assigned the same value when the function is called.

Add the correct return type for the function:

```
double feet_and_inches_to_meters(int feet, double inches) {

  double meters = (feet + inches/12) / 3.28;

  std::cout << feet << " feet and " << inches << " inches is:\n";
  std::cout << meters << " meters!\n";

  return meters;

}
```

You got it!

Put together the function:

```
void cat_sounds( int times ) {

  for (int i = times; i > 0; i--) {

    std::cout << "Meow! ";

  }

}
```

You got it!

What function does every C++ program have?

`main()`

 `main()` is necessary for a C++ program to run!

`iostream`

`rand()`

If you have several functions in your program, where is a good spot to put the declarations?

Below `main()`.

Inside of `main()`.

In a header file.

 The header file is often named **main.hpp**.

What's the problem with this function?

```
say_hi() {

  std::cout << "Hi!";

}
```

There's no return type.

 All C++ functions require a return type. In this case, it should be `void`.