Review Questions:

1. Approximately what is the line #include <stdio.h> at the top of a C source file for?
2. What are some uses for comments?
3. Why is indentation important? How carefully does the compiler pay attention to it?
4. What are the largest and smallest values that can be reliably stored in a variable of type int?
5. What is the difference between the constants 7, '7', and "7"?
6. What is the difference between the constants 123 and "123"?
7. What is the function of the semicolon in a C statement?

Exercises:

1. Get the “Hello, world!" program to work on your computer.

2. What do these loops print?

```
for(i = 0; i < 10; i = i + 2)
        printf("%d\n", i);

for(i = 100; i >= 0; i = i - 7)
        printf("%d\n", i);

for(i = 1; i <= 10; i = i + 1)
        printf("%d\n", i);

for(i = 2; i < 100; i = i * 2)
        printf("%d\n", i);
```

3. Write a program to print the numbers from 1 to 10 and their squares:

```
1       1
2       4
3       9
...
10      100
```

4. Write a program to print this triangle:

```
*
**
***
****
*****
******
*******
********
*********
**********
```

Don't use 10 printf statements; use two nested loops instead. You'll have to use braces around the body of the outer loop if it contains multiple statements:

```
for(i = 1; i <= 10; i = i + 1)
        {
        /* multiple statements */
        /* can go in here */
        }
```

(Hint: a string you hand to printf does not have to contain the newline character \n.)

Assignment #1 ANSWERS

Question 1. ***Approximately what is the line #include <stdio.h> at the top of a C source file for?***

In the case of our first few programs, it lets the compiler know some important information about the library function, printf. It also lets the compiler know similar information about other functions in the ``Standard I/O library,'' some of which we'll be learning about later. (It also provides a few I/O-related definitions which *we'll* be using later.) We'll learn more about the #include directive when we cover the C Preprocessor in a few weeks.

Question 2. ***What are some uses for comments?***

Describing what a particular variable is for, describing what a function is for and how it works, documenting the name, version number, purpose, and programmer of an entire program, explaining any tricky or hard-to-understand part about a program.

Question 3. ***Why is indentation important? How carefully does the compiler pay attention to it?***

Indentation is important to show the logical structure of source code, in particular, to show which parts--the ``bodies`` of loops, if/else statements, and other control flow constructs--are subsidiary to which other parts.

Indentation is *not* specially observed by the compiler; it treats all ``white space'' the same. Code which is improperly indented will give a human reader a mistaken impression of the structure of the code, an impression potentially completely different from the compiler's. Therefore, it's important that the punctuation which describes the block structure of code to the compiler matches the indentation.

Question 4. ***What are the largest and smallest values that can be reliably stored in a variable of type int?***

32,767 and -32,767.

Question 5. ***What is the difference between the constants 7, '7', and "7"?***

The constant 7 is the integer 7 (the number you get when you add 3 and 4). The constant '7' is a character constant consisting of the character `7' (the key between `6' and `8' on the keyboard, which also has a `&' on it on mine). The constant "7" is a string constant consisting of one character, the character `7'.

Question 7. ***What is the function of the semicolon in a C statement?***

It is the statement terminator.

Exercise 3. *Write a program to print the numbers from 1 to 10 and their squares.*

```
#include <stdio.h>

int main()
{
	int i;

	for(i = 1; i <= 10; i = i + 1)
		printf("%d  %d\n", i, i * i);

	return 0;
}
```

Exercise 4. *Write a program to print a simple triangle of asterisks.*

```
#include <stdio.h>

int main()
{
	int i, j;

	for(i = 1; i <= 10; i = i + 1)
		{
		for(j = 1; j <= i; j = j + 1)
			printf("*");
		printf("\n");
		}

	return 0;
}
```