Programming in C

Lecture #5
Kenneth W. Flynn
RPI CS

Outline
- Theory: Scope
- Application: Scope in C
- Exercise: Activity 5.1
- Application: Data Types
- Application: ASCII Characters
- Exercise: Activity 5.2

Theory
- Scope is a somewhat tricky topic in programming languages
- In general, scope refers to where and when a variable exists

Scope in C
- Scope in C only depends on where a variable is, not when
- A variable is defined for the extent of the block in which it is declared
Scope Example

```c
int main (void)
{
    int a;
    /* Stuff */
    return 0;
}
```

Tell-e-Scope

- The simplest case of scope is for a variable defined at the beginning of a function.
- It is in scope throughout that function, until the `}` which ends it.
- Remember you can only declare variables at the start of a function.
- Things get more complicated with parameters to functions, however

Scope Example II

```c
int square (int x);

int main (void) int square (int x)
{
    int q = 3, r;
    int sq;
    r = square (q);
    sq = x*x;
    return sq;
}
```

Scope Example Explained

- Main has two variables in scope, q and r
- Square has two variables in scope, sq and x
- The x in square is a parameter (since it is passed to the function)
- As far as scope is concerned, it is a new variable which is in scope for the duration of the function

Actual & Formal Parameters

- Formal parameters: The variables which receive the arguments
  - In our example, x is a formal parameter
- Actual parameters: The parameters passed to the function
  - In our example, q is a actual parameter
- When a function is called, the value of each of the actual parameters is determined, and copied into the new formal parameter, which comes into scope

When A Function Is Called...

- So, q = 3
- When square (q) is called, the value of q (3) is copied into x
- The function is then evaluated; at the end, all variables are discarded, so the x goes "poof"
- The return value is then copied into r
Name Conflicts
- When a function’s formal parameters or local variables share the name with variables anywhere else in your program, they are distinct entities
  - Different values
  - Different scope
- You can only access the formal parameters or local variables inside a function, so you really don’t have to worry about this

Globals
- NO!
- That said, global variables are variables defined outside of any function. Their scope is the entire program
- Don’t use global variables. C provides mechanisms such that they are almost never useful, and certainly NEVER in this course

More Function Practice
- Write a function that takes two parameters -- the number of dice to roll, and the number of sides those dice have. It should return the total of rolling the dice.
- Write a program that reads a number of dice and sides, and calls the above function 10 times, and computes the average value rolled.

Data Types
- We’ve seen two examples so far: int and float
- There are several more; let’s examine them

Activity 5.1
Data Type Values

<table>
<thead>
<tr>
<th></th>
<th>Signed</th>
<th>Unsigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>char</td>
<td>-128..127</td>
<td></td>
</tr>
<tr>
<td>short</td>
<td>-32768..32767</td>
<td>0..65536</td>
</tr>
<tr>
<td>int</td>
<td>Varies</td>
<td>Varies</td>
</tr>
<tr>
<td>long</td>
<td>-2147483648..2147483647</td>
<td>0..4294967295</td>
</tr>
</tbody>
</table>

/* Examples */
unsigned long q;
short c;

Notes On Types
- We can use these types just like any others, including the "unsigned" types. We'll see an application of unsigned types next week.
- Format specifiers (printf, scanf): %ld
  %hd -- long and short, respectively
- Characters: %c

Characters
- Characters are a special data type. Although they are also integers from -128..127, they also represent the letters on the keyboard, and other symbols.
- For example, 'A' is 65
- This direct mapping is called the ASCII character set. p. 891 of the text.

You're Such a Character!
- In our code, we don't have to look up the values in the ASCII table all the time.
- We can use character constants, like 'A' or '&' just like we use 0, 1 or 42.
- We can declare variables of type char, read keypresses into them, and then do things based on the character typed.

Character Practice
- Write a program that reads a character from the user and prints out its ASCII value. Write a function to do this conversion. What should that function return?
- Write a program to simulate a calculator: read two integers and then an operation: +, -, *, /, or %. Then perform that operation on those numbers. You might want to use a switch statement.