Variables in C

CMSC 104 Section 02
Feb 21, 2024

Administrative Notes

- We’ll go over quiz 1 in class today; if you have any questions see me after class

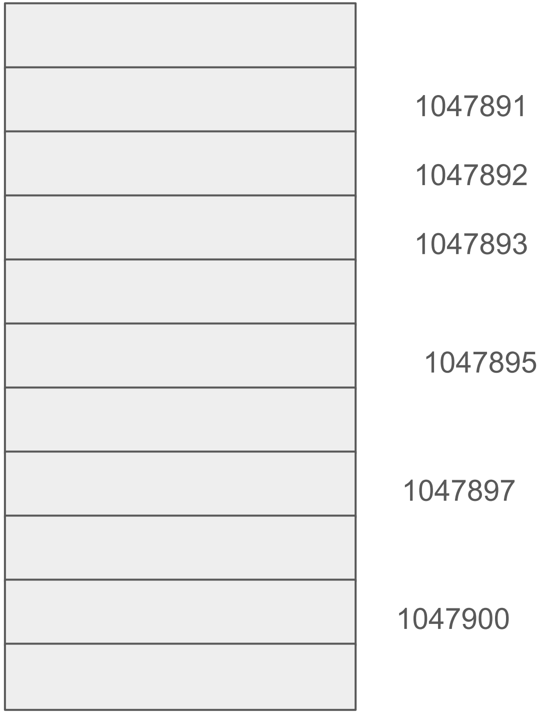
A quick reset - the program from Monday

```
//This is a c program that prompts the user for two integers,
/* reads them in,
   adds them,
   and then prints out the result
*/

#include <stdio.h>
int main ()
{
  int first_integer;  // the first integer the user will type in
  int second_integer; // the second integer the user will type in
  printf("This program will read in two integers, add them and print the result\n");

  printf("Please enter the first integer \n");  //prompt for the first integer
  scanf("%d", &first_integer);  // read the first integer

  printf("Please enter the second integer \n"); //prompt for the second integer
  scanf("%d", &second_integer); //read the second integer

  int sum;   //the variable where we will store the sum
  sum = first_integer + second_integer;

  printf("The sum of your numbers is %d\n", sum); //print the sum

}
```

How memory works in a computer

Memory ("RAM") can be thought of as a long, one-dimensional array of cells that store strings of 1's and 0's - "bits".

Each location in memory has an address

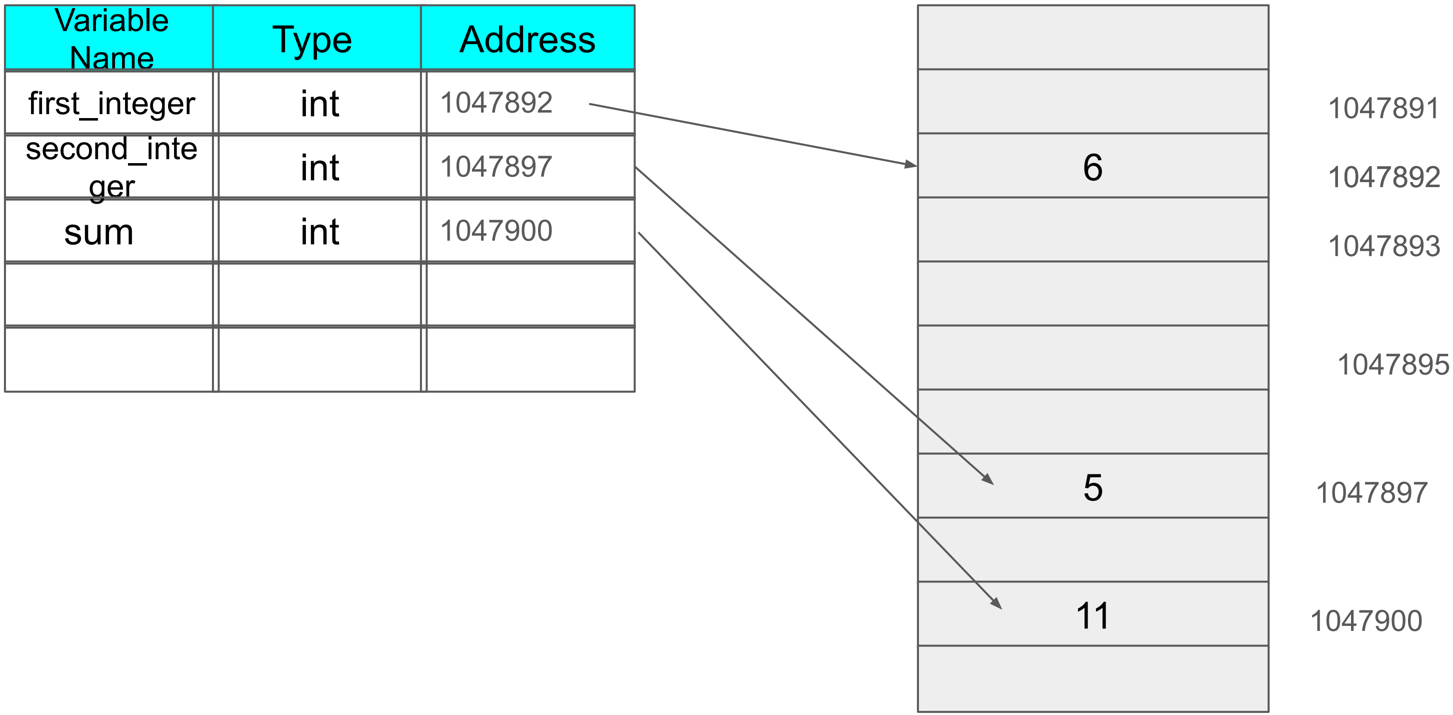


A symbol table

C, like many languages, creates a “symbol table” for each part of a program

- It contains all variables defined for the program, the type of the variable, and a pointer to the address in memory where the value of that variable is defined

Symbol table



Variable Name	Type	Address
first_integer	int	1047892
second_integer	int	1047897
sum	int	1047900

How Variables work in C

This stuff is pretty standard in all high-level computer languages; let’s look at how C is unique

Rules for variable names

- Can contain letters, digits & underscore
- Can't start with a digit
- Can't be an existing "reserved word"
- Are case sensitive - "Grade" is not the same variable as "grade"
- Can be as long as you want but only the first 31 characters in the variable name are important.
    - "abcdefghijklmnopqrstuvwxyz0123456789" and "abcdefghijklmnopqrstuvwxyz0123456777" are the same variable

C reserved words (“keywords”)

These words cannot be used as variable names in C

`auto`	`break`	`int`	`long`
`case`	`char`	`register`	`return`
`const`	`continue`	`short`	`signed`
`default`	`do`	`sizeof`	`static`
`double`	`else`	`struct`	`switch`
`enum`	`extern`	`typedef`	`union`
`float`	`for`	`unsigned`	`void`
`goto`	`if`	`volatile`	`while`

CMSC104 Naming Conventions

These are not rules enforced by C, but they are conventions used in our class to make your code easier to read:

- Begin variable names with lowercase letters.
- Use meaningful identifiers (name)
- Separate “words” within identifiers with underscores or mixed upper and lower case.
  - Example: surfaceArea, surface_area, surface_Area
  - Be consistent!

To emphasize:

Legal identifiers refer to the restrictions C places on naming identifiers, such as not beginning with a number.

Naming conventions refer to the standards you must follow for this course, such as all variable names must be lowercase. But this isn’t technically required by the compiler.

A quick check

Which of the following variable names are legal for C:

AREA

3D

lucky***

num45

Last-Chance

#values

x

yt3

pi

num$

%done

area_under_the_curve

Another check:

Which of the following are legal C variable names, but do not follow CMSC104 Naming Conventions?

Area

person1

Last_Chance

values

x

yt3

pi

finaltotal

numChildren

area_under_the_curve

Basic types: int, float, char

(we’ll add more types later on in the semester)

C has three basic predefined data types:

- Integers (whole numbers)
    - int, long int (or just long), short int (or just short)
    - Always positive: unsigned (unsigned int, unsigned long, unsigned short)
- Floating point (real numbers)
    - float, double
- Characters
    - char

But they’re all bits

It’s just a string of 1’s and 0’s

- you can do things with a variable that you might not have originally intended

Examples

What if the variable was an int, and you entered a character?

What if the variable was a float, and you entered an integer?

What if the variable was a char, and you entered a float?

Another example

```
#include <stdio.h>

int main() {
float fathoms;// the depth in fathoms
float feet;// the depth in feet
float inches; //the depth in inches

/* Ask the user for the depth in fathoms */
printf(“Enter the depth in fathoms: “)
scanf(“%f”, &fathoms);
/* Convert depth from fathoms to inches */
feet = 6.0 * fathoms
inches = 12.0 * feet

/* Display The Results */
printf ( ”Its depth at sea:\n” );
printf( ”\t%1.2f fathoms\n” , fathoms);
printf( ”\t%1.2f feet\n”, feet);
printf (”\t%1.2f inches\n”, inches);
return 0 ; }
```