

C programming language

Jeremy Iverson

College of Saint Benedict & Saint John's University

- Dennis Ritchie and Brian Kernighan creators of C circa 1972
- TODO: more thorough history



Dennis Ritchie in 2011 / CC BY 2.0



Brian Kernighan in 2012 / CC BY 2.0

comparison

Java	C
object-oriented	procedural
interpreted	compiled
String	char array
condition (boolean)	condition (int)
garbage-collected	no memory management
references	pointers
exceptions	error codes

- in Java, everything is a method that is called on an object
- in C, everything is a function
- in Java, source code is compiled to byte code, which is then interpreted by Java VM
- in C, source code is compiled into binary machine code
- in Java, String is a class
- in C, a string is just an array of **char** values which ends with the **char** `'\0'`
- in Java, the Java VM takes care of deallocating memory used
- in C, any memory you allocate, you must also deallocate

hello, world

```
1 /* file: helloworld.c */  
2  
3 #include <stdio.h>  
4  
5 int main() {  
6     printf("hello, world\n");  
7     return 0;  
8 }
```

```
$ gcc -o helloworld helloworld.c  
$ ./helloworld  
hello, world
```

- The tradition of using the phrase "Hello, world!" as a test message was influenced by an example program in the seminal book *The C Programming Language*

variables

```
1 // file: figure2-4.c
2 // Stan Warford
3 // A nonsense program to illustrate global variables
4
5 #include <stdio.h>
6
7 char ch;
8 int j;
9
10 int main() {
11     scanf("%c %d", &ch, &j);
12     j += 5;
13     ch++;
14     printf("%c\n%d\n", ch, j);
15     return 0;
16 }
```

```
$ gcc -o figure2-4 figure2-4.c
$ ./figure2-4
M 419
N
424
```

- What *kind* of variables are `ch` and `j`?
- What would you expect for input '~ 2147483643'?
- This program would behave exactly the same had this program declared `ch` and `j` as local variables instead of global variables.

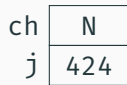
memory model — part i

global variables

declared outside of any function and remain in place throughout the execution of the entire program. they are stored at a fixed location in memory.

local variables

declared within a function and come into existence when the function is called and cease to exist when the function terminates. they are stored on the run-time stack.

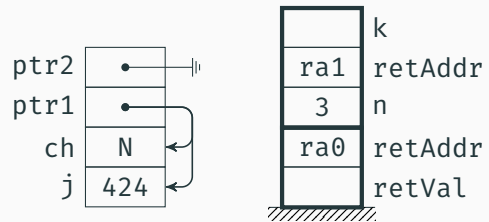


(a) Fixed location.



(b) Run-time stack.

- (a) and (b) represent the state of relevant memory for the previous program just before it terminates, i.e., in the process of executing line 15.



(a) Fixed location.

(b) Run-time stack.

conditions

- under what conditions will each of the following be execute?

```
1 if (x) {  
2     /* ??? */  
3 }  
4 if (x-y) {  
5     /* ??? */  
6 }  
7 if (x=y) {  
8     /* ??? */  
9 }
```

- $x \neq 0$

- $x \neq y$

- $y \neq 0$



except where otherwise noted, this worked is licensed under creative commons attribution-sharealike 4.0 international license