

Computer Science 50

Introduction to Computer Science I

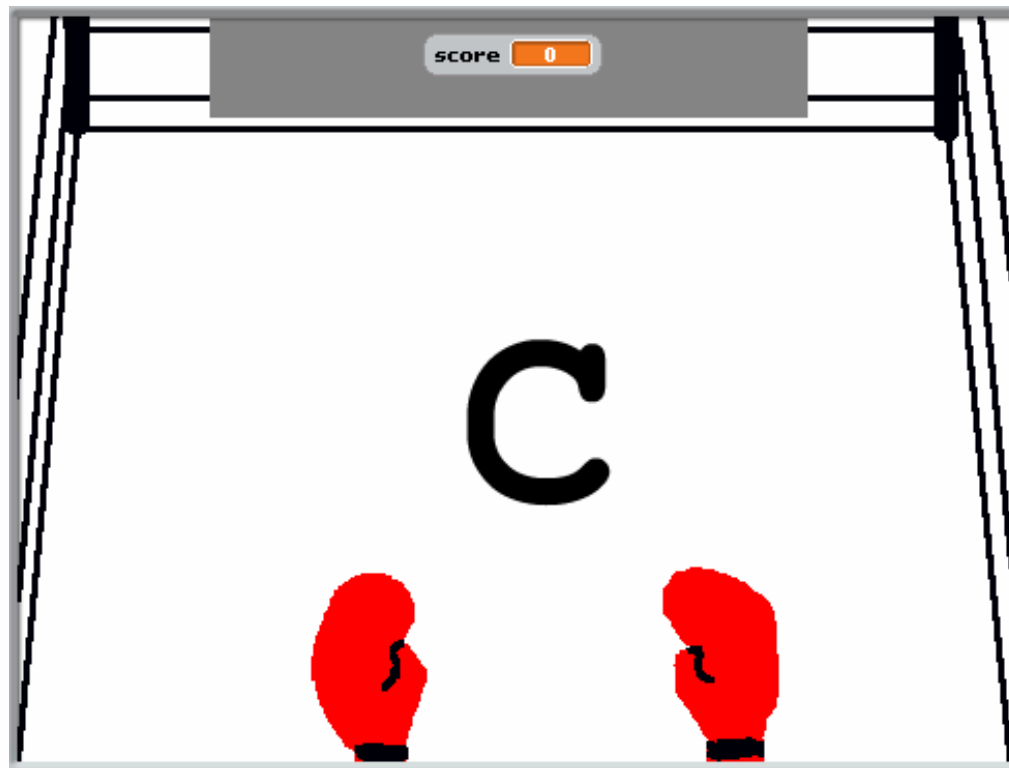
Harvard College

Week 1

David J. Malan

malan@post.harvard.edu

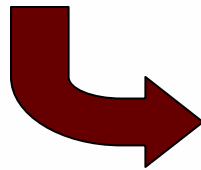
Scratch versus C



Hello, C!

```
#include <stdio.h>

int
main(int argc, char * argv[])
{
    printf("hello, world\n");
}
```

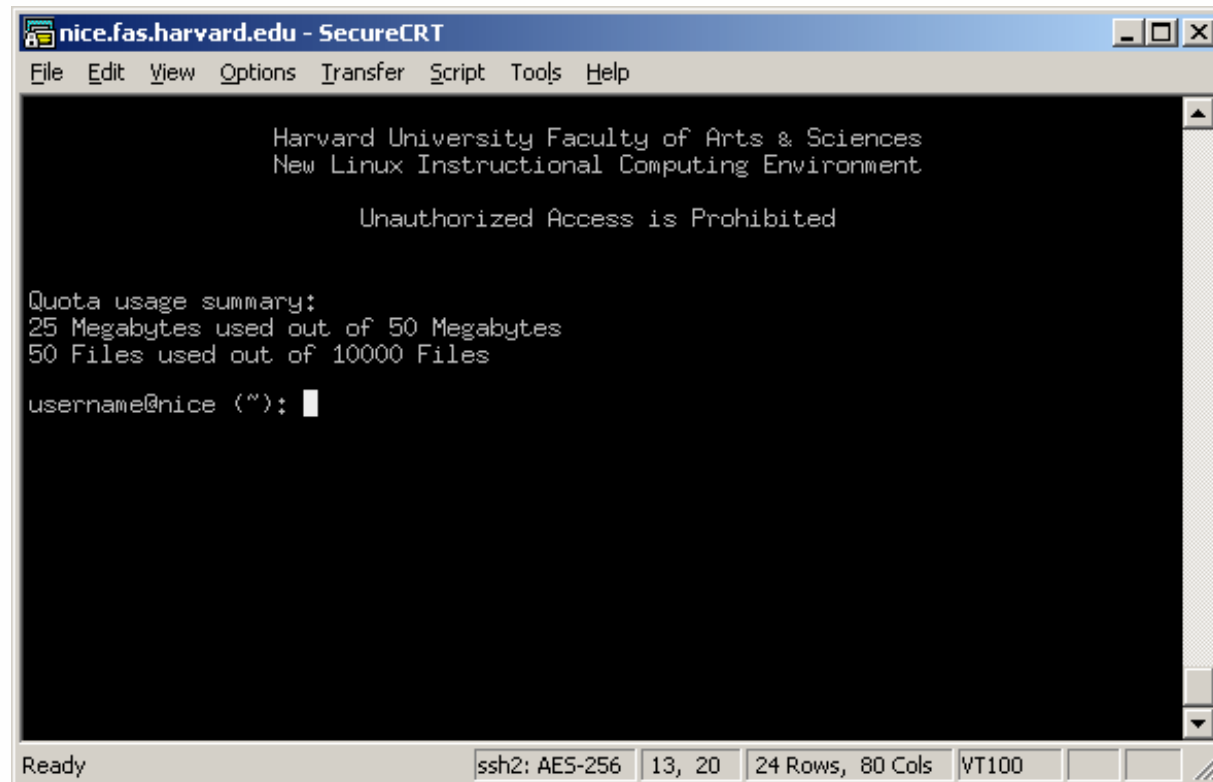


```
10000011 00000001 00010001 00000000 00111101 11111100 01110100 00111101
00000000 01000000 00000000 00000000 00000000 00000000 00000000 00000000
10010000 00000000 00000000 00000000 01010000 00000000 00000111 00110000
00001011 00000001 00001011 00000011 00001010 00000000 00000000 00000000
00000000 00100000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00100000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
01110000 00010000 00000000 00100000 00000001 00000000 00000000 00000000
00000000 00000000 00000000 00100000 00000001 00000000 00000000 00000000
00000000 00000000 00000000 01000000 00000001 00000000 00000000 00000000
00000000 00100000 00000000 01000000 00000001 00000000 00000000 00000000
11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
10010000 10000000 00000000 01000000 00000001 00000000 00000000 00000000
00101110 01100100 01111001 01101110 01100001 01101101 01101001 01100011
10110000 00000100 00000000 00100000 00000001 00000000 00000000 00000000
10110000 00000100 00000000 00100000 00000001 00000000 00000000 00000000
10100000 00000001 00000000 00000000 00000000 00000000 00000000 00000000
10110000 00000100 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00100000 00000000 00000000
[...]
```

nice.fas.harvard.edu



nice.fas.harvard.edu

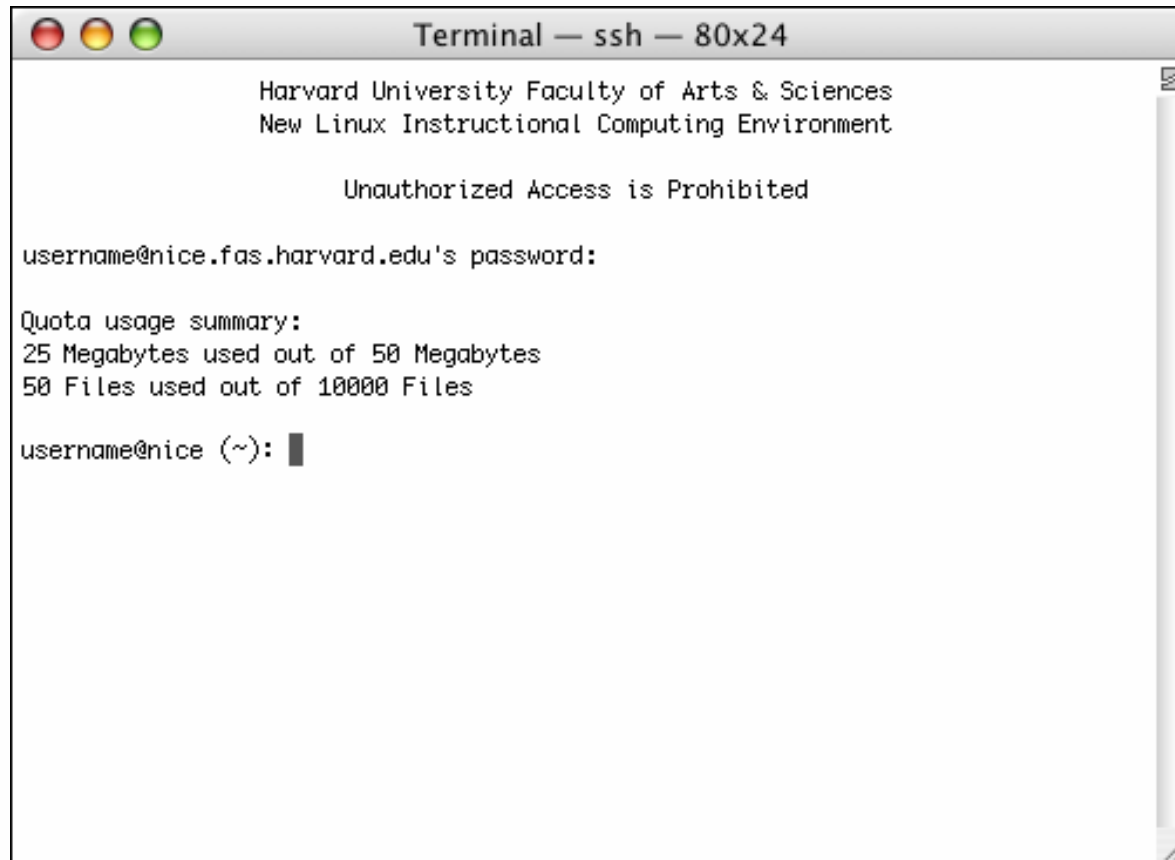


The image shows a SecureCRT terminal window titled "nice.fas.harvard.edu - SecureCRT". The window has a menu bar with "File", "Edit", "View", "Options", "Transfer", "Script", "Tools", and "Help". The terminal content is as follows:

```
Harvard University Faculty of Arts & Sciences  
New Linux Instructional Computing Environment  
  
Unauthorized Access is Prohibited  
  
Quota usage summary:  
25 Megabytes used out of 50 Megabytes  
50 Files used out of 10000 Files  
  
username@nice (~):
```

The status bar at the bottom of the window displays "Ready", "ssh2: AES-256", "13, 20", "24 Rows, 80 Cols", and "VT100".

nice.fas.harvard.edu



```
Terminal — ssh — 80x24
Harvard University Faculty of Arts & Sciences
New Linux Instructional Computing Environment

Unauthorized Access is Prohibited

username@nice.fas.harvard.edu's password:

Quota usage summary:
25 Megabytes used out of 50 Megabytes
50 Files used out of 10000 Files

username@nice (~):
```

Some Commands

- :: `cd`
- :: `cp`
- :: `ls`
- :: `mkdir`
- :: `mv`
- :: `pwd`
- :: `rm`

Some More Commands

```
:: ci
:: co
:: cs50submit
:: emacs
:: gcc
:: gdb
:: make
:: man
:: nano
:: sb
:: vi
```


How to Write a Program in C

- 1) `nano hello1.c`
- 2) `gcc hello1.c`
- 3) `a.out`

see
`hello1.c`

How to Write a Program in C

(with a better name)

- 1) `nano hello1.c`
- 2) `gcc -o hello1 hello1.c`
- 3) `hello1`

see
`hello1.c`

main

```
int main(int argc, char * argv[]);
```

Standard Output

`printf`

```
int printf(const char *format, ...);
```

see

<http://www.cppreference.com/stdio/printf.html>

Escape Sequences

- :: `\n`
- :: `\r`
- :: `\t`
- :: `\"`
- :: `\\`

see

http://www.cppreference.com/escape_sequences.html

Variables

Types †

- :: `char`
- :: `double`
- :: `float`
- :: `int`

† `long`, `short`, `signed`, `unsigned`

see
`math1.c`

Format Strings

:: **%c**

:: **%d**

:: **%e**

:: **%E**

:: **%f**

:: **%s**

:: **%u**

:: **%x**

see
math2.c, sizeof.c
<http://www.cppreference.com/stdio/printf.html>

Arithmetic Operators

⋮ +

⋮ -

⋮ *

⋮ /

⋮ %

Precedence

Operator	Description	Associativity
() [] . -> ++ --	Parentheses (grouping) Brackets (array subscript) Member selection via object name Member selection via pointer Postfix increment/decrement (see Note 1)	left-to-right
++ -- + - ! ~ (type) * & sizeof	Prefix increment/decrement Unary plus/minus Logical negation/bitwise complement Cast (change <i>type</i>) Dereference Address Determine size in bytes	right-to-left
* / %	Multiplication/division/modulus	left-to-right
+ -	Addition/subtraction	left-to-right
<< >>	Bitwise shift left, Bitwise shift right	left-to-right
< <= > >=	Relational less than/less than or equal to Relational greater than/greater than or equal to	left-to-right
== !=	Relational is equal to/is not equal to	left-to-right
&	Bitwise AND	left-to-right
^	Bitwise exclusive OR	left-to-right
	Bitwise inclusive OR	left-to-right
&&	Logical AND	left-to-right
	Logical OR	left-to-right
?:	Ternary conditional	right-to-left
= += -= *= /= %= &= ^= = <<= >>=	Assignment Addition/subtraction assignment Multiplication/division assignment Modulus/bitwise AND assignment Bitwise exclusive/inclusive OR assignment Bitwise shift left/right assignment	right-to-left
,	Comma (separate expressions)	left-to-right

Width and Precision †

```
:: %<width>.<precision>e  
:: %<width>.<precision>E  
:: %<width>.<precision>f  
:: %<width>.<precision>s
```

† -, +

see

math3.c

<http://www.cppreference.com/stdio/printf.html>

Variables

Types †

:: `bool`

:: `string`

† These are CS 50-specific.

see
`hello2.c`

Standard Input †

```
:: char GetChar();  
:: double GetDouble();  
:: float GetFloat();  
:: int GetInt();  
:: string GetString();
```

† These are CS 50-specific.

see
hello3.c, adder.c

How to Write a Program in C

(using CS 50's library)

- 1) `nano hello3.c`
- 2) `gcc -o hello3 hello3.c -lcs50`
- 3) `hello3`

see
`hello3.c`

Fahrenheit to Celsius

$$C = (5/9) \times (F - 32)$$

```
#include <cs50.h>
#include <stdio.h>

int
main(int argc, char * argv[])
{

}

}
```

Conditions

`if`

```
if (condition)
{
    /* do this */
}
```

Conditions

if-else

```
if (condition)
{
    /* do this */
}
else
{
    /* do that */
}
```

see
conditions1.c

Conditions

if-else if-else

```
if (condition)
{
    /* do this */
}
else if (condition)
{
    /* do that */
}
else
{
    /* do this other thing */
}
```

see
conditions2.c

Boolean Expressions

```
if (condition || condition)
{
    /* do this */
}
else
{
    /* do that */
}
```

Boolean Expressions

```
if (condition && condition)
{
    /* do this */
}
else
{
    /* do that */
}
```

see
nonswitch.c

Conditions

switch

```
switch (expression)
{
    case i:
        /* do this */
        break;
    case j:
        /* do that */
        break;
    default:
        /* do this other thing */
}
```

see
switch{1,2}.c

Loops

for

```
for (initializations; condition; updates)
{
    /* do this again and again */
}
```

see
`progress{1,2}.c`

Loops

while

```
while (condition)
{
    /* do this again and again */
}
```

see
progress3.c

Loops

do-while

```
do
{
    /* do this again and again */
}
while (condition)
```

see
positive.c

Computer Science 50

Introduction to Computer Science I

Harvard College

Week 1

David J. Malan

malan@post.harvard.edu