



COMP 125: Basics of Computer Organization

Week 04, Lecture 06



Agenda

1. Review: Variables, functions
2. The math we need
3. Practice!

Variables and Functions

Why?

- Readability
- Reusability

Types

- Numbers
- Strings
- Boolean (true/false)
- Code (function)

What can we *do* with our data?

- Compare it
- Do math on it
- Display it
- Change it
- Delete it
- Make copies of it

Arithmetic (math basics)

- “Operator”
 - The thing we’re doing
- “Operands”
 - The things we’re doing the operator *on*
- Order of operations:
 - PEMMDAS
- Left-associated
- Return a number

Op	Definition
+	Add
-	Subtract
*	Multiply
/	Divide
%	Modulus (remainder)
**	Exponent

Comparators

- Return True/False
- Parenthesis *matter*

Op	Meaning
>	Greater than
<	Less than
==	Equal to
!=	Not equal to
>=	Greater than or equal to
<=	Less than or equal to

Variables

“=” means “Assignment”

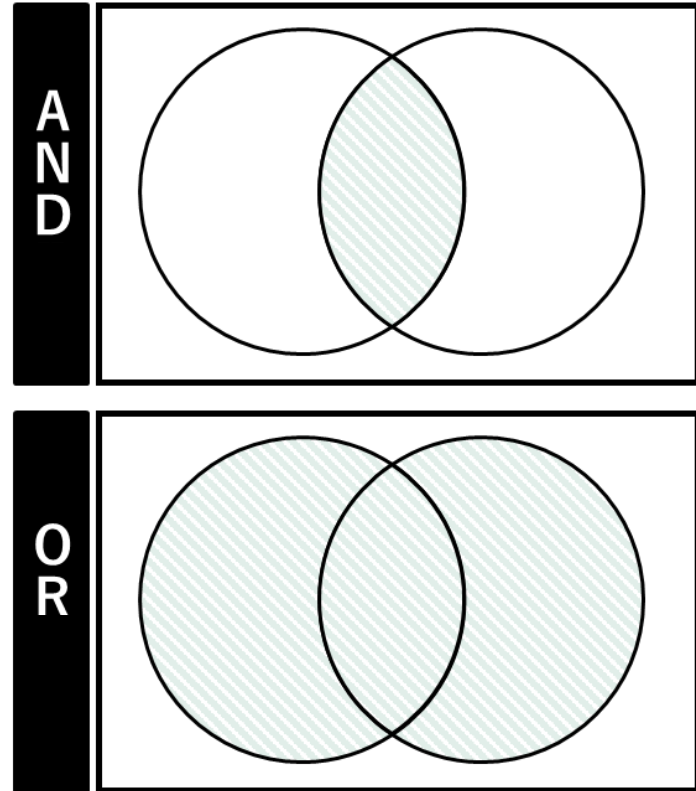
This is important.

= does *not* compare equality.

Comparators and Boolean

- Return true/false
- `true == 1`
- `false == 0`

Op	Meaning
<code>&&</code>	AND (both True)
<code> </code>	OR (either is True)
<code>!</code>	NOT



Board examples:

```
a = 5
b = 10
c = -10
d = 10
e = 1
g = 2
h = 0
```

```
t = true
f = false
```

```
a + b
a * b
b ** a
d / g - e
d / (g - e)
```

```
a > b
b > c
(a > b) > g
(a > b) > c
```

```
h = h
h == h
g >= a
a <= c
d == c
```

```
h == f
e == t
t > f
t >= f
t > g
f > g
```

```
t && t
t || t
t || f
t && f
f && t
t && f ||
t
t && f ||
f
!t
!f
!t == f
!f > 0
```

Controlling Our Code

Need to be able to say “if this is true, do this, otherwise, do that”

Need to be able to say “repeat this task until this thing happens”

Why?

- Computers should do work for us
- Can do more complicated things
- Solve bigger problems

Controlling Our Code

Need to be able to say “if this is true, do this, otherwise, do that”

```
if (this thing){  
    then do this other thing  
}  
else if (this other thing) {  
    then go do this  
}  
else {  
    do the last-ditch thing  
}
```

Controlling Our Code

Need to be able to say “repeat this task until this thing happens”

```
while (this thing is still true){  
    then do this thing  
}
```

Controlling Our Code

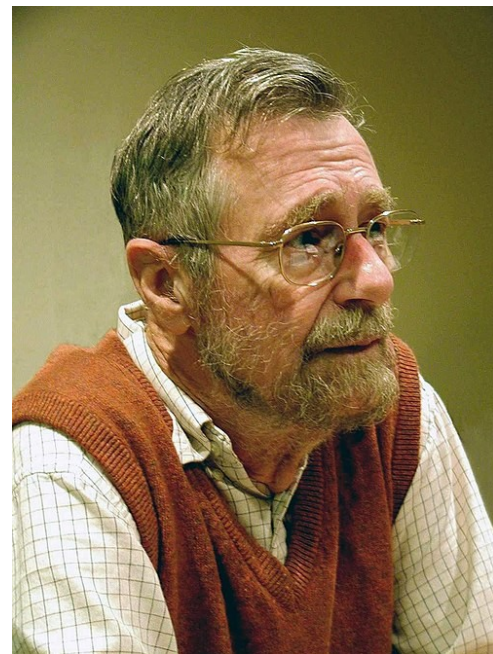
Need to be able to say “repeat this task until this thing happens”

```
for (something that starts here and goes until this){  
    do this thing  
}
```

Controlling Our Code

Solving a problem with code: the top-down approach

1. Begin with the problem
2. Solve it in the most generic way possible
3. Specify one non-specific part
4. Repeat until you have something that works!



Edsger W. Dijkstra

Father of computer
programming

Controlling Our Code - Exercise

Count from 0 to 100

Controlling Our Code - Exercise

Count from 1 to 100

Simplest!

Show on the screen the numbers from 1 to 100

Controlling Our Code - Exercise

Count from 0 to 100

Simplest!

```
text(1, 10, 10)
```

```
text(2, 10, 15)
```

```
text(3, 10, 20)
```

```
...
```

Controlling Our Code - Exercise

Count from 0 to 100

Store the number in a variable

```
var i = 1
```

```
text(i, 10, 10)
```

```
i = 2
```

```
text(i, 10, 20)
```

```
i = 3
```

```
text(i, 10, 30)
```

```
...
```

Controlling Our Code - Exercise

Count from 0 to 100

Store the number in a variable

```
var i = 1
```

```
text(i, 10, 10)
```

```
i = i + 1
```

```
text(i, 10, 20)
```

```
i = i + 1
```

```
text(i, 10, 30)
```

```
...
```

Controlling Our Code - Exercise

Count from 0 to 100

Now we have reusability!

```
var i = 1
```

```
text(i, 10, 10)
```

```
i = i + 1
```

```
text(i, 10, 20)
```

```
i = i + 1
```

```
text(i, 10, 30)
```

```
...
```

Controlling Our Code - Exercise

Count from 1 to 100

Now we have reusability!

```
var i = 1
```

Do the following from 1 to 100:

```
text(i, 10, 10)
```

Controlling Our Code - Exercise

Count from 1 to 100

Now we have reusability!

```
var i = 1
```

Do the following from 1 to 100:

```
text(i, 10, 10)
```

```
i = i + 1
```

Controlling Our Code - Exercise

Count from 0 to 100

Store the number in a variable

```
var i = 1
```

```
text(i, 10, 10)
```

```
i = 2
```

```
text(i, 10, 20)
```

```
i = 3
```

```
text(i, 10, 30)
```

```
...
```

Controlling Our Code - Exercise

Count from 1 to 100

Now we have reusability!

```
var i = 1
```

Do the following from 1 to 100:

```
text(i, 10, (10*i))
```

```
i = i + 1
```


Controlling Our Code - Exercise

Count from 1 to 100

while:

```
var i = 1
```

Do the following from 1 to 100:

```
text(i, 10, (10*i))
```

```
i = i + 1
```

Controlling Our Code - Exercise

Count from 1 to 100

while:

```
var i = 1
```

```
while (i <= 100){
```

```
    text(i, 10, (10*i))
```

```
    i = i + 1
```

```
}
```

Controlling Our Code - Exercise

Count from 1 to 100

while:

```
var i = 1  
while (i <= 100){  
    text(i, 10, (10*i))  
    i = i + 1  
}
```

Controlling Our Code - Exercise

Count from 1 to 100

for:

```
for (var i = 1; i <= 100; i = i + 1){  
    text(i, 10, (10*i))  
}
```