COMP 150:
Basics of Computer Organization
Week 04, Lecture 06

Agenda

1. Conditionals and Loops
2. Controlling Our Code

Review: Boolean Logic

Exercises

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

elif (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
```

Another conditional:

Try - catch

- Reads like “try this, and if that doesn’t work, do that”

It’s a way to handle errors

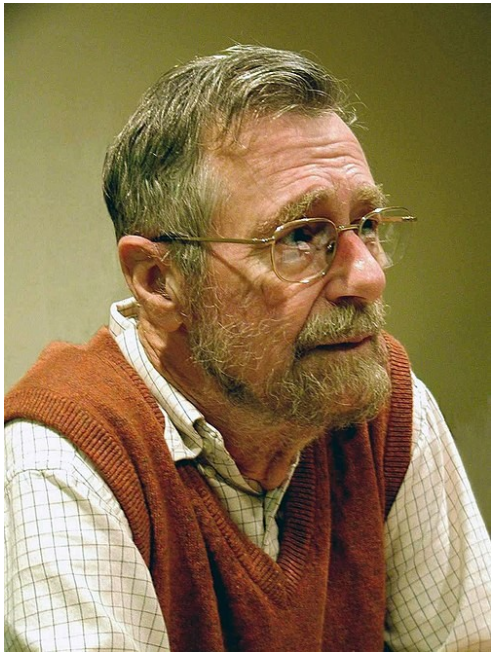
Controlling Our Code

Exercises

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 1 to 100

Simplest!

```
Print the numbers from 1 to 100
```

Controlling Our Code - Exercise

Count from 1 to 100

Simplest!

```
print(1)
print(2)
print(3)
...
```

Controlling Our Code - Exercise

Count from 1 to 100

Store the number in a variable

```
i = 1

print(i)

i = 2

print(i)

i = 3

print(i)

…
```

Controlling Our Code - Exercise

Count from 1 to 100

Store the number in a variable

```
var i = 1

print(i)

i = i + 1

print(i)

i = i + 1

print(i)

…
```

Controlling Our Code - Exercise

Count from 1 to 100

Now we have reusability!

```
var i = 1

Do the following from 1 to 100:

print(i)
```

Controlling Our Code - Exercise

Count from 1 to 100

Now we have reusability!

```
var i = 1
Do the following from 1 to 100:
print(i)
i = i + 1
```

Controlling Our Code - Exercise

Count from 1 to 100

while:

```
var i = 1
Do the following from 1 to 100:
print(i)
i = i + 1
```

Controlling Our Code - Exercise

Count from 1 to 100

while:

```
var i = 1
while (i <= 100):
    print(i)
   i = i + 1
```

Controlling Our Code - Exercise

Count from 1 to 100

while:

```
var i = 1
while (i <= 100):
    print(i)
   i = i + 1
```

Controlling Our Code - Exercise

Count from 1 to 100

for:

```
for i in range(0, 101):
    print(i)
```

Controlling Our Code - Exercise

We want a program that prints the average of a dataset. We know there will always be five elements in the dataset.

This time, our data is: [3, 7, 11, 13, 17].

Controlling Our Code - Exercise

We want a program that prints the average of a dataset. We know there will always be five elements in the dataset.

This time, our data is: 3, 7, 11, 13, 17

```
Give the average of the dataset.
```

Controlling Our Code - Exercise

We want a program that prints the average of a dataset. We know there will always be five elements in the dataset.

This time, our data is: 3, 7, 11, 13, 17

```
Give the average of the numbers 3, 7, 11, 13, and 17.
```

Controlling Our Code - Exercise

We want a program that prints the average of a dataset. We know there will always be five elements in the dataset.

This time, our data is: 3, 7, 11, 13, 17

```
Print the average of the numbers 3, 7, 11, 13, and 17.
```

Controlling Our Code - Exercise

We want a program that prints the average of a dataset. We know there will always be five elements in the dataset.

This time, our data is: 3, 7, 11, 13, 17

```
Add the numbers 3, 7, 11, 13, and 17, then divide by 5, and
print the value.
```

Controlling Our Code - Exercise

We want a program that prints the average of a dataset. We know there will always be five elements in the dataset.

This time, our data is: 3, 7, 11, 13, 17

```
l = [3, 7, 11, 13, 17]
```

```
Add the contents of l, then divide by 5, and print the value.
```

Controlling Our Code - Exercise

We want a program that prints the average of a dataset. We know there will always be five elements in the dataset.

This time, our data is: 3, 7, 11, 13, 17

```
l = [3, 7, 11, 13, 17]
sum = 0
for x in l:
        sum = sum + x
Then divide sum by 5, and print the value.
```

Controlling Our Code - Exercise

We want a program that prints the average of a dataset. We know there will always be five elements in the dataset.

This time, our data is: 3, 7, 11, 13, 17

```
l = [3, 7, 11, 13, 17]
sum = 0
for x in l:
        sum = sum + l
result = sum / 5
print(result)
```

Controlling Our Code - Exercise

We want a program that prints the average of a dataset. We know there will always be five elements in the dataset.

This time, our data is: 3, 7, 11, 13, 17

```
l = [3, 7, 11, 13, 17]

sum = sum(l)

result = sum / 5

print(result)
```

Controlling Our Code - Exercise

We want a program that prints the average of a dataset. We **do not know** how many elements there will be in the dataset.

This time, our data is: 3, 7, 11, 13, 17

```
l = [3, 7, 11, 13, 17]

sum = sum(l)

result = sum / 5

print(result)
```

Controlling Our Code - Exercise

We want a program that prints the average of a dataset. We **do not know** how many elements there will be in the dataset.

This time, our data is: 3, 7, 11, 13, 17

```
l = [3, 7, 11, 13, 17]

sum = sum(l)

result = sum / the number of elements in l

print(result)
```

Controlling Our Code - Exercise

We want a program that prints the average of a dataset. We **do not know** how many elements there will be in the dataset.

This time, our data is: 3, 7, 11, 13, 17

```
l = [3, 7, 11, 13, 17]

sum = sum(l)

number_in_l = 0

for x in l:

       number_in_l = number_in_l + 1

result = sum / number_in_l

print(result)
```

Controlling Our Code - Exercise

We want a program that prints the average of a dataset. We **do not know** how many elements there will be in the dataset.

This time, our data is: 3, 7, 11, 13, 17

```
l = [3, 7, 11, 13, 17]
sum = sum(l)
number_in_l = len(l)
result = sum / number_in_l
print(result)
```

Controlling Our Code - Exercise

We want a program that 1. understands assignment grades, 2. shows us how many students received each grade, and 3. gives statistics on the assignment scores

We have the following grades on an assignment:

- Four students earned 90/100
- Three students earned 85/100
- Two students earned 70/100
- Two students earned 55/100