Why programming?

Virtually everything in the modern world has a software (program) component.

Programming is an essential skill in many scientific areas.
Programming in the Real-World

- ocean modeling
- diffusion MRI of brain
- food web in Serengeti
- nuclear physics
- colliding galaxies
- an aerosol droplet containing coronavirus
- airflow over landing gear
What will you learn?

1. How to create simple programs
2. How to learn more on your own
Class logistics

For:
- Announcements
- Questions after class
- Posting class recordings

Weekly homeworks out after each class
- Due Friday of the following week
- Posted on the website
- Try to finish homeworks on time, but if you are stuck, ask for help on piazza!
- *Optional* small class project later (early June). You can work in groups for that

Please ask questions during class! Send private messages through zoom or unmute yourselves!

Website: https://www.cs.princeton.edu/~pparedes/teaching/mise/summer23/
What is a program?

Collection of well defined instructions that describe a task

To bake a cake:

1. Combine sugar and butter
2. Beat in eggs
3. Add flour
4. Add milk
5. Bake in oven
How do we tell a computer what to do?

To communicate amongst ourselves we use human languages (for example, English).

To communicate with computers we will use a programming language, namely **Python**.
What does a python program look like?

Suppose we want to write a program that checks if a number is prime:

Look for all numbers between 2 and target
If we find a divisor:
Then the number isn’t prime, report that
Otherwise:
The number is prime! Report that

```
num = 407
for i in range(2, num):
    if (num % i) == 0:
        print(num, "is not a prime number")
        break
else:
    print(num, "is a prime number")
```
Program Abstraction

Input Stream → Python Program

input() → print() → Output Stream
First demo: my first program

Go to: https://www.programiz.com/python-programming/online-compiler/

Learning goals:

1. How to print/read data
2. Python as a calculator
3. What is a variable
Pop Quiz 1:

What is the output of the following program:

```python
1  a = "Hello World"
2
3  print("a")
```
What we just learned

- Printing text to the shell
- What is a variable, how to create modify and use one
- Reading input from the user in the shell
- How to use basic math operators
You have to follow the right syntax!

Here is an example of wrong code:

```python
1 a = "Hello World"
2
3 print(a")
```

Produces an error:

```
File "<string>", line 3
   print(a")
     ^
SyntaxError: EOL while scanning string literal
```

To fix we need to change line 3.
Basic Data Types in Python

- **Integer** - represents integers eg. 2, -20, 9999999
- **Float** - represents decimal numbers eg. 2.0, -3.99, 568.98
- **String** - represents a sequence of characters eg. “Hello”, “#Hi!”, “42”
  - Note the pair of double quotes around the sequence
- **Boolean** - represents true or false values eg. True, False
  - Capitalization is important here!
Second demo: examples with data types

Learning goals:

1. What is a data type
2. How to use strings
3. How to manipulate data types
## The `type()` instruction!

If you ever need to know the type of something, you can use the `type()` instruction to do that, like so:

```python
main.py

print("Hello world")
print(type("Hello world"))
print(5)
print(type(5))
print(5.0)
print(type(5.0))
print(True)
print(type(True))
```

<table>
<thead>
<tr>
<th>main.py</th>
<th>Shell</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 print(&quot;Hello world&quot;)</td>
<td>Hello world</td>
</tr>
<tr>
<td>2 print(type(&quot;Hello world&quot;))</td>
<td>&lt;class 'str'&gt;</td>
</tr>
<tr>
<td>3 print(5)</td>
<td>5</td>
</tr>
<tr>
<td>4 print(type(5))</td>
<td>&lt;class 'int'&gt;</td>
</tr>
<tr>
<td>5 print(5.0)</td>
<td>5.0</td>
</tr>
<tr>
<td>6 print(type(5.0))</td>
<td>&lt;class 'float'&gt;</td>
</tr>
<tr>
<td>7 print(True)</td>
<td>True</td>
</tr>
<tr>
<td>8 print(type(True))</td>
<td>&lt;class 'bool'&gt;</td>
</tr>
</tbody>
</table>
Second demo continued

print(2+3*4)  # prints 14, not 20
print(5+4%3)  # prints 6, not 0 (% has same precedence as *, /, and //)
print(2**3*4) # prints 32, not 4096 (** has higher precedence than *, /, //, and %)

print(5-4-3)  # prints -2, not 4 (- associates left-to-right)
print(4**3**2) # prints 262144, not 4096 (** associates right-to-left)
Pop Quiz 2:

What is the output of the following program:

```plaintext
1 a = 5
2 b = 10
3 print(a / 2)
4 a = 3
5 print(b // 9)
```
Pop Quiz 3:

What is the output of the following program:

```python
1  a = "Hello "
2  print(a * 3 + "World!")
```
One more demo: A polite program

Learning goals:

1. Writing a program with a purpose

```python
1   name = input()
2
3   print("Welcome to programming, " + name + ")")
```
Pop Quiz 4:

If we run this program and write the number "3" on the terminal, what would be its output?

```python
1  n = input()
2  print("Thanks!" * n)
```
One more demo: A polite program now complete

Learning goals:

1. Writing a program with a purpose

```python
name = input()
print("Welcome to programming, " + name + "!")
print("Pick a number:")
n = int(input())
print("Thanks!" * n)
```
Homeworks - Using CodeForces

CodeForces

A website where you can submit programs and get immediate feedback

Go to: https://codeforces.com/group/K1Fxw6skwV/contests
What's next?

Homework will be posted on Piazza by tomorrow! You won’t learn anything if you don’t try the homeworks.

Class 2: Functions and Conditionals

How to create code modules
How to have different outcomes based on the input