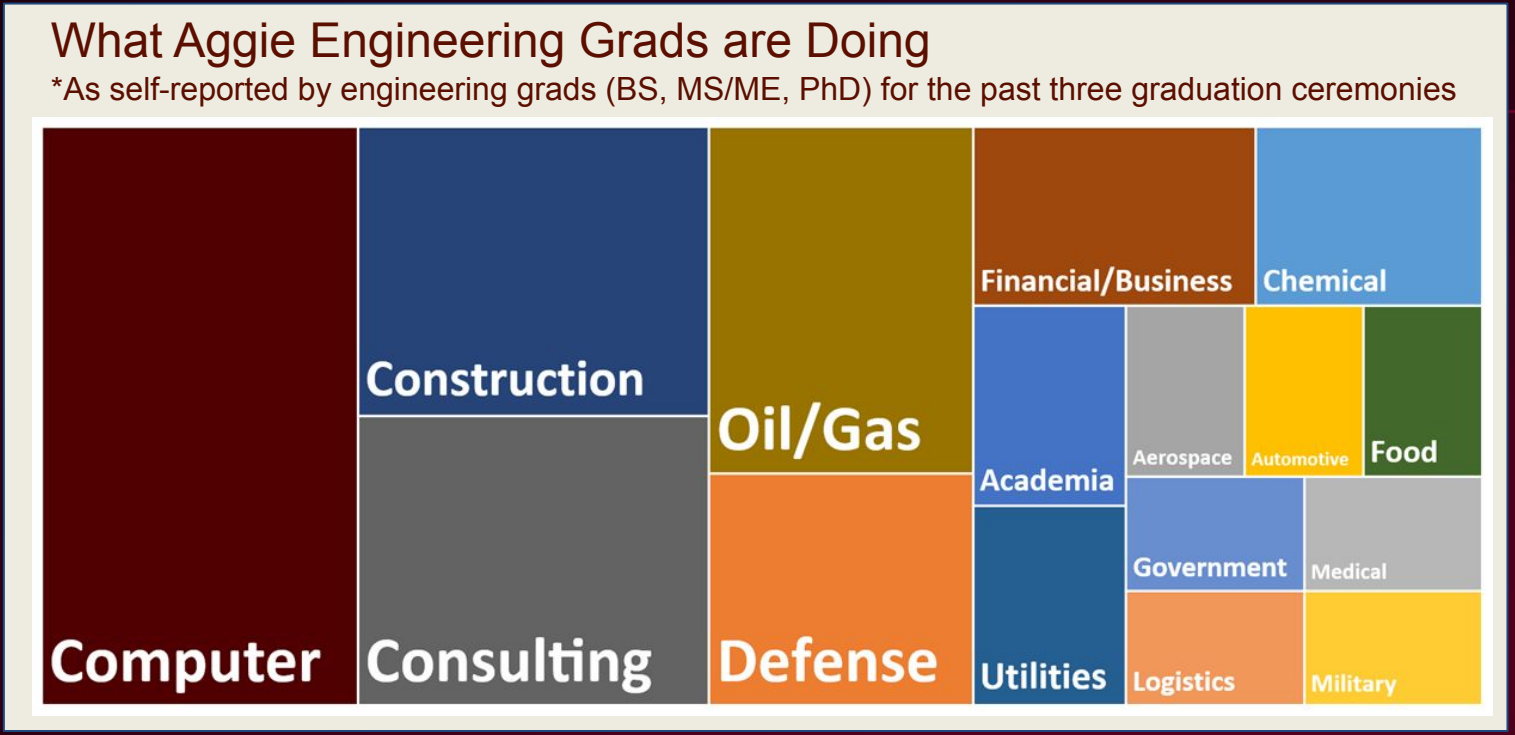


ENGINEERING
TEXAS A&M UNIVERSITY

Meeting 1

Introduction to Course, Welcome to Engineering, What is Programming and Why is this the first thing being covered in Engineering?

ENGINEERING
TEXAS A&M UNIVERSITY

Welcome to ENGR 102

Mr. Craig Spears

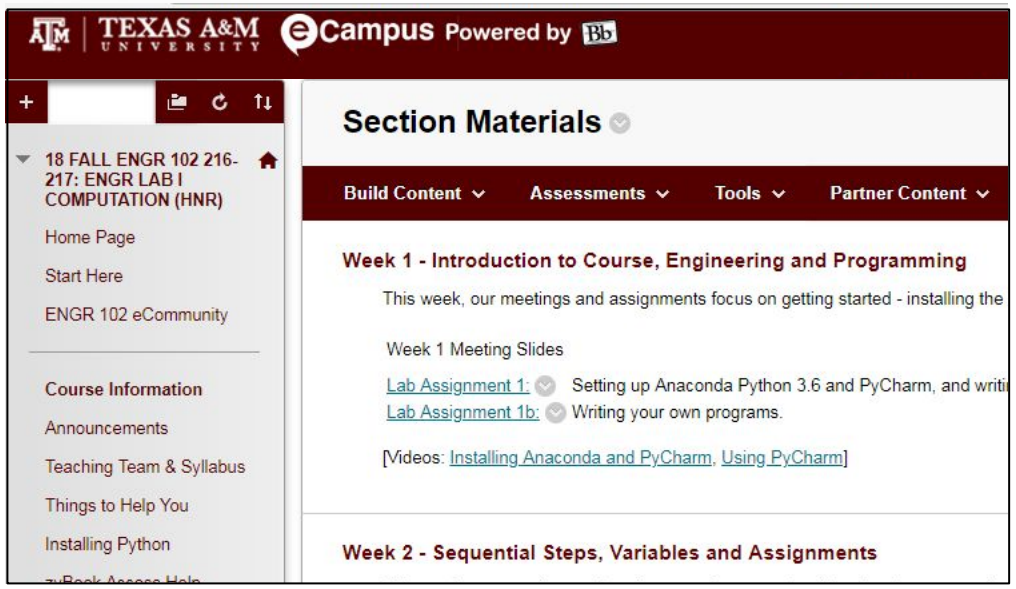
Craig.Spears@tamu.edu

Office: ZACH 482-D

- Bachelor and Master's degrees in ChemE from A&M
- Taught chemistry at Blinn
- Worked at Bryan Research & Engineering 10+ years
  - Chemical Engineering Consultants
  - Sr. Technical Support & Sr. Consulting Engineer
- Started teaching ENGR courses in 2016
- Associate Professor of the Practice

eCampus & Syllabus

eCampus



Syllabus

- course description
- expectations
- book
- grading
- calendar

Classroom Rules

No watching Netflix, ESPN videos, etc… (it's distracting)

- I may ask you to leave - you will not be allowed to make up any missed work.

Safe place to learn

- Learn through practice - fail, repeat, improve.
- Plan to fail and learn! We're here to challenge you, not to see what you already know.
- Do not interrupt others, laugh at others, etc.

Challenge yourself

- Be involved - don't back off and "manage" your team.

Piazza

Online forum for questions and discussions

Only email me when the answer only applies to you personally - otherwise post on Piazza for everyone to benefit (anonymous or not)

Written Assignments

Engineering is a profession that places great importance on the ability to clearly and effectively communicate ideas and results.

- No matter how efficient the design or how great the analysis happens to be, the information needed to describe it must be conveyed in a clear manner
- Lettering that is easily read and understood still remains a tool that is required of today's engineers

Work where the lettering / writing is unreadable or hard to decipher will be considered wrong.

ENGINEERING
TEXAS A&M UNIVERSITY

Academic Integrity

"An Aggie does not lie, cheat, or steal or tolerate those who do."

please visit aggiehonor.tamu.edu

Academic Integrity

- Knowing and applying correct knowledge of citation and documentation standards (learn what plagiarism entails).
    - library.tamu.edu is a great source for more information
- Utilizing materials and resources that are approved by the instructor.
- Understanding when collaboration is allowed, and with whom.
- Not short-changing yourself on your education, and ability to succeed in the future.

Your Time Expectations (in class)

ENGR 102 is a 2-credit class:

- 1 hour per week of “lecture” (1 credit) = instructor-driven
- 3 hours per week of “lab” (1 credit) = hands-on work (programming, teamwork, quizzes, exams)

Attendance is required

- Quizzes and exams during meeting times
- Lab work will involve working in teams
- Attendance will affect lab grade

Your Time Expectations (outside class)

In college, you should expect to spend 2-3 hours outside of class *per credit hour* on the subject, weekly.

- So, for ENGR 102, expect to spend 4-6 hours a week outside the classroom

You should expect to spend time on:

- College of Engineering Modules (0.5 hours/week)
- Reading (zyBook) (1 hour/week)
- Lab assignments (3-5 hours/week)
- Studying (regularly)

Questions to Ask Yourself

- What do I know about these engineering majors?
- What courses will I take to earn a degree in this major?
- Am I enjoying my courses? Do I feel challenged or stressed? (Grades are NOT everything!)
- Am I willing to spend the time it takes to complete this degree?
- What kinds of jobs will this major prepare me for? Which sounds most interesting?
- What kinds of skills will I need to do the job I want? Where can I get them?

Grand Challenges

National Academy of Engineering (NAE) 14 Grand Challenges for Engineering

http://www.engineeringchallenges.org/

- Make Solar Energy Economical
- Provide Energy from Fusion
- Develop Carbon Sequestration Methods
- Manage the Nitrogen Cycle
- Provide Access to Clean Water
- Restore and Improve Urban Infrastructure
- Advance Health Informatics
- Engineer Better Medicines
- Reverse-Engineer the Brain
- Prevent Nuclear Terror
- Secure Cyberspace
- Enhance Virtual Reality
- Advance Personalized Learning
- Engineer the Tools of Scientific Discovery

Why Are We Studying Programming?

Computer programs are a fundamental tool across all of engineering

- Computing will be used in your classes (especially engineering classes)
- Computing will be used at your job
  - using programs
  - modifying programs
  - writing programs
- Even if you use computing programs only as a tool, understanding how your tools work will make you more proficient with them!

Why Are We Studying Programming?

Computing is transforming all of engineering (and every other field)

- Over the last 20 years, computing has revolutionized most of society:
  - internet
  - social networks
  - portable/ubiquitous computing (smartphones)
  - data analytics
  - simulation
  - etc.
- Many recent advancements and research directions closely tie in to computing

Why Are We Studying Programming?

For some of you, computing will be the central focus of your study

- Computer Science and Computer Engineering especially
- Several other majors will require additional computing, often as a fundamental aspect of the field
- This material will provide a basis for further more advanced studies

Why Are We Studying Programming?

Computer programming is an element of engineering on its own

- Involves designing and building a solution to solve some problem, or enable something new to be done
- Producing software is similar to building other things
  - Design
  - Analysis
  - Construction
  - Testing
- The “material” used to build with is computer code
- Not all engineering involves physical devices
  - Some fields focus on processes, for instance!

Recent Survey

In a recent survey of Texas A&M Engineering graduates, asking them what their job involved, Computing was listed more frequently than every other topic!

As we prepare you for being engineering leaders for the next decades, computing skills will be a critical component of your skill set.

First week's activities/goals

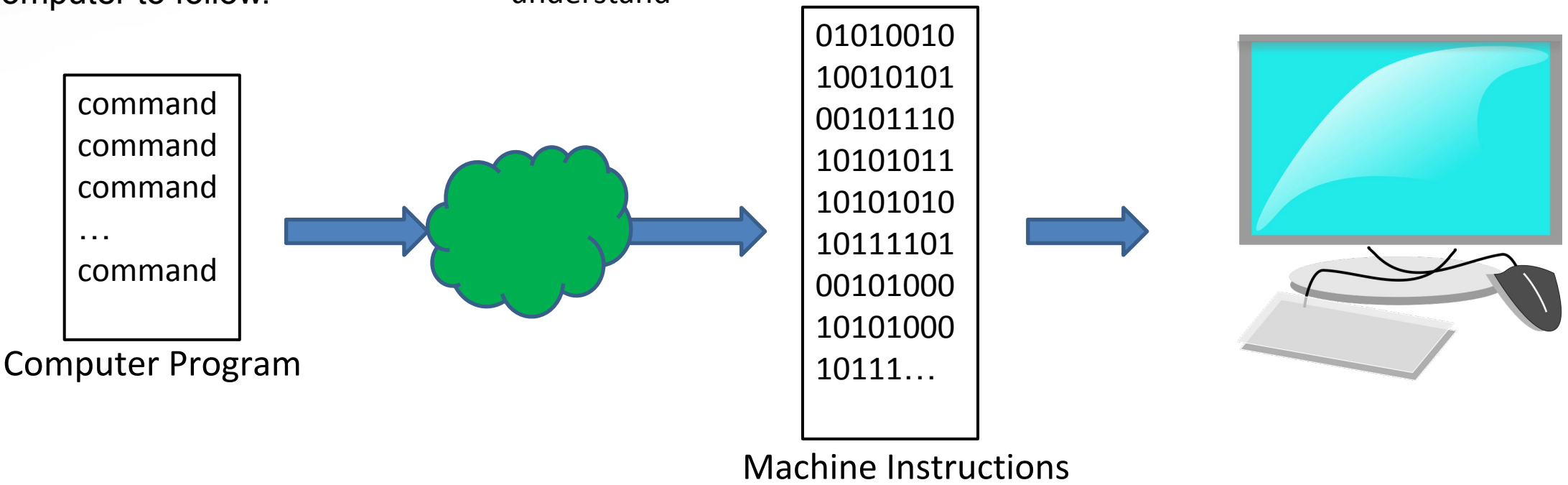
- Install the Python Compiler (Anaconda / Python)
- Install the IDE (PyCharm)
- Write your first "Howdy, World!" program
- Practice computing mathematical values

Programming

Background, and why Python?

Computer Programming – The Basic Process

1. A programmer writes commands / instructions, called a **program**, for the computer to follow.
2. A **compiler** or **interpreter** takes those commands and converts them to instructions that the computer can understand
3. When a program is **run**, the instructions are **executed** on the computer



The Program

- A program is just a text file with computer commands in it.
- You can, generally, use any editor you want to write a program
  - Saving as a text file, with the right extension: .py for python, for instance
- Some editors are designed to help write programs
  - Can color-code text, give hints, autofill, etc.
- Often, an editor is tied together with a compiler and other features into an Integrated Development Environment (IDE)
  - We will be using an IDE for this course

Compiler/Interpreter

- The compiler or interpreter translates the program into machine instructions.
- Think of a compiler as pulling all the information in the program together, then generating machine instructions.
  - Since it can look at the whole program, it can possibly create more optimal machine instructions
  - The machine instructions are typically saved to a file
- Think of an interpreter as translating line by line.
  - Often, the execution happens as it is interpreted, so there is no intermediate file saved
- Python is usually thought of as interpreted.

Execution

- For a compiled program, the execution involves “running” the program.
  - Most applications and programs you would run are compiled programs, ready to be executed
- For an interpreted program, execution often happens right as the program is interpreted
  - So, the “program” that is run is just the original program – the machine instructions might not be saved to a separate file
  - Sometimes, the original program is called a “script”
- With an IDE, the execution often takes place in a window or part of the IDE itself

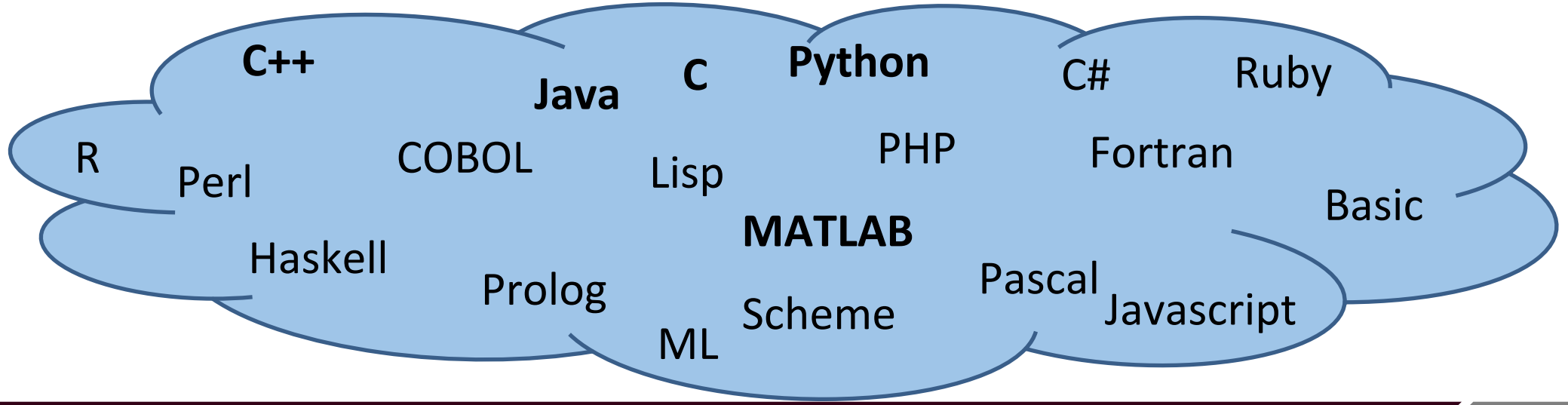
Programming Languages

- The computer processor understands commands in binary.
  - It is incredibly painful for people to write significant amounts of code in binary
- Assembly language provides a direct mapping between the binary commands and something a person can read
  - Essentially, it just translates from binary to a text representation - still very difficult to write
- Programming languages have been developed to help *people* understand the instructions given to a computer.
  - Although they are still commands to the computer, they are conceptually easier to understand
  - It is usually much easier to comprehend program ideas/purpose in a programming language
  - It is usually more efficient: one statement in a programming language can replace many assembly/machine instructions

Programming Languages

Many languages have been developed.

- Each has its own benefits and drawbacks
- They have increased and decreased in popularity over time
- There is no “best” language, though some languages are better than others for particular tasks



Why Python?

Our main focus in this class will be on Python programming. Python is generally considered:

- Easy to learn
- Powerful (real applications are written with it)
- Flexible (can be used for many different applications – lots of library support)
- Popular (it is being used more and more)

Random (foot?) note: Python gets its name from Monty Python, the British comedy troupe!

Getting Set Up

In the lab portion, you will work on getting the environment set up, and writing your first programs. What you'll need:

- A Python interpreter (Python – Anaconda Distribution)
- A Python IDE (PyCharm)

We will be using the IDE to develop our programs, compile them, and execute them.

- But, if you just save your .py files on your computer, and have the Python interpreter installed, you can execute them directly

Meeting 2

What is Engineering? What is Programming

Learning Objectives

Introduction to Programming

- Define the terms IDE, Compiler, and Interpreter
- Use the print statement in Python
- Define the purpose of, and utilize in a program, the import function in Python (e.g., from math import *)
- Use simple mathematical operations in Python (+, -, *, /, //, %, **, sqrt, cos, sin, tan, log, log10, exp)
- Use comments in a Python program

Your First Program

- We will develop more advanced programs as the course goes on, but will start out with some basic ones.
- The traditional first program people write is called “Hello, World”
  - We’ll adapt it slightly, but importantly.
- We can do this with one line of code:

```
print("Howdy, World!")
```

The print statement

Program

```
print("Howdy, World!")

print(1)
```

Output

```
Howdy, World!
1
```

The print command instructs the computer to print whatever is in the parentheses to the screen

Numerical values can be placed directly

For text, the text to print should be in quotation marks

Printing

- Each print statement will print one value on one line
- We can put multiple print statements into one program

```
print("Why do gorillas have big nostrils?")
print("Big fingers!")
```

- We'll see more about how to use print statements later in the course
  - But, feel free to experiment, or learn on your own!

Note: "print" is a generic term for "display" (usually to the screen), in contexts like this. It does not mean "print on paper using a printer."

Mathematical Calculations

- Mathematical computations are one of the most common things we'll do in this class (and a core part of almost all programs)
- Python has support for the basic arithmetic operations

Addition	+
Subtraction	-
Multiplication	*
Division	/

- Order of operations is enforced `2+3*4 = 14`
- Parentheses are also enforced `(2+3)*4 = 20`

More Mathematical operations (built in)

Power: **

```
2**10 = 1024
```

Integer Division (division without remainder): //

```
7//3 = 2
```

Modulo (remainder from division - result is also called modulus): %

```
7%3 = 1
100%10 = 0
```

More mathematical functions

- We will add one line to our programs to give us access to additional mathematical functions.
  - We'll see exactly what this is doing later on
- For now, put the following line at the top of your program:

```
from math import *
```

- Mathematical functions will have a name and parentheses that enclose the value to take the function of

  e.g. `cos(0)` is the cosine of 0

Common mathematical functions

Remember to start program with: `from math import *`

- `sqrt(x)` = square root of x
- `cos(x)`, `sin(x)`, `tan(x)` = trigonometric functions
- `acos(x)`, `asin(x)`, `atan(x)` = inverse trigonometric functions
- `log(x)`, `log10(x)` = logarithm, base e or base 10
- `exp(x)` = e^x (e is the base for natural logarithm)

There are many more…

Note: trig functions use radians, not degrees

What would the following program print?

```
from math import *
print("ENGR")
print(((3**3)+4*5)*(sqrt(25)//2)+(28%10))
```

What would the following program print?

```
from math import *
print("ENGR")
print(((3**3)+4*5)*(sqrt(25)//2)+(28%10))

Output:
ENGR
102.0
```

Why?

```
((3**3)+4*5)*(sqrt(25)//2)+(28%10)
(( 27 )+ 20)*(   5     //2)+(  8  )
(      47   )*(       2    )+(  8  )
(           94              )+(  8  )
                     102
```

Why is 102.0 printed instead of 102?
We'll see in 1-2 weeks.

A comment about comments

- Code can also include “comments”
- These start with a # character
  - Everything on the line after the # is ignored by the computer
- Comments are there to help users
  - We’ll come back to these in more detail, later
- For now, we will use comments to write information at the top of our program, for class purposes.
  - To record your name, etc.

Current Assignments

ENGINEERING
TEXAS A&M UNIVERSITY

Complete the CHEN module on the eCommunity page

- These will be available only for their designed week, so don't delay
- Due 9/1, by end-of-day

Lab Assignment 1	Due 9/1, by end-of-day *(11:58 PM or earlier)*
Lab Assignment 1b	Due 9/1, by end-of-day

Preactivity: Complete zyBook chapters 1 & 2 prior to class next week