

MCS-177
Introduction to
Computer Science I

Louis Lei Yu

Objectives

- To understand how computers can help solve real problems
- To further explore numeric expressions, variables, and assignment
- To understand the accumulator pattern
- To utilize the math library
- To further explore simple iteration patterns
- To understand simple selection statements
- To use random numbers to approximate an area

1

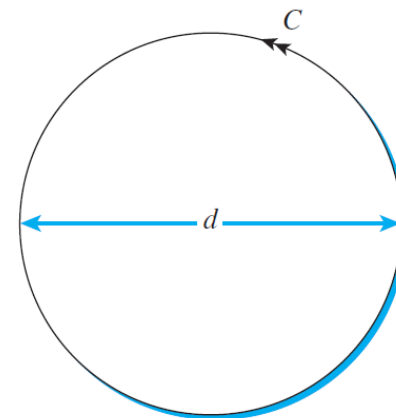
What is PI?

Ratio of circumference to diameter

3.1415926535897932384626433832795028841
9716939937510...

`math.pi` from the `math` module

Figure 2.1



```

>>> import math
>>> help (math)
Help on module math:

NAME
  math

MODULE REFERENCE
  http://docs.python.org/3.3/library/math

  The following documentation is automatically generated from the Python
  source files. It may be incomplete, incorrect or include features that
  are considered implementation detail and may vary between Python
  implementations. When in doubt, consult the module reference at the
  location listed above.

DESCRIPTION
  This module is always available. It provides access to the
  mathematical functions defined by the C standard.

FUNCTIONS
  acos(...)
    acos(x)

    Return the arc cosine (measured in radians) of x.

  acosh(...)
    acosh(x)

    Return the hyperbolic arc cosine (measured in radians) of x.

  asin(...)
    asin(x)

    Return the arc sine (measured in radians) of x.

  asinh(...)
    asinh(x)

```

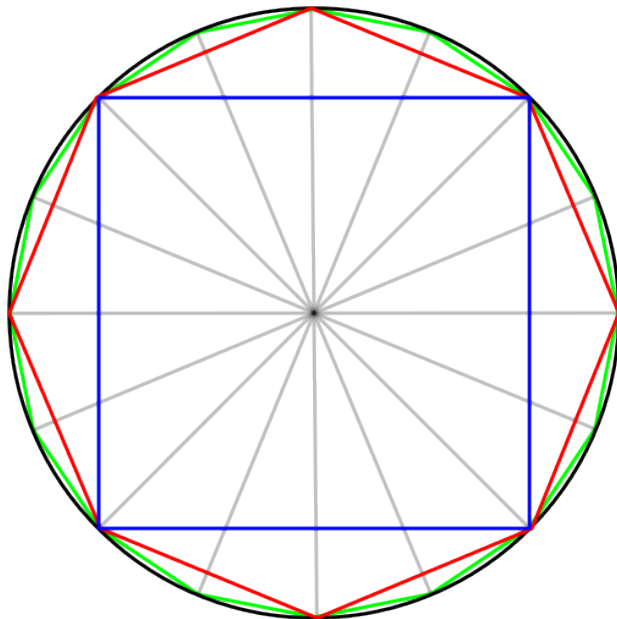
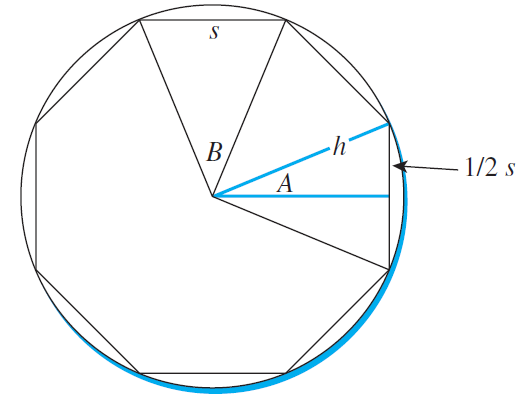
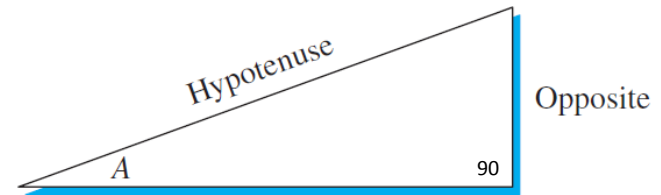


Figure 2.3



$$\sin A = \frac{\text{Opposite}}{\text{Hypotenuse}}$$

Let's code this:

- From the math module, we need:

- math.sin
- math.radians

```
>>> help(math.sin)
Help on built-in function sin in module math:

sin(...)
    sin(x)

    Return the sine of x (measured in radians).
```

```
>>> help(math.radians)
Help on built-in function radians in module math:

radians(...)
    radians(x)

    Convert angle x from degrees to radians.
```

```
>>> import math
>>> sides = 8
>>> angleB = 360.0/sides
>>> angleA = angleB/2
>>> opposite = math.sin(math.radians(angleA))
>>> base = 2 * opposite
>>> polygonCircumference = base * sides
>>> pi = polygonCircumference / 2
>>> pi
3.0614674589207183
```

Template for function definition with return

```
def functionName(param1,param2,...):
    statement1
    statement2
    ...
    return expression
```



The rest...

```
>>> archimedes
<function archimedes at 0x103ee8a70>
>>> archimedes(8)
3.0614674589207183
>>> archimedes(16)
3.121445152258052
>>> archimedes(100)
3.141075907812829
>>> archimedes(1000)
3.141587485879563
```

- The print statement
- print vs return
- Calling function inside a function