CSI 33 Lecture 3

We will review:

- Classes in Python and C++

Classes in Python

- Python is object-based, *classes* can be implemented
- *classes* have *attributes* and *behaviors(methods, functions)*
- *objects* are *instances of classes*
- *classes* can be *inherited* from (*is-a* relationship)
- each method definition has one required argument, usually called *self*

```
class Dice(GeometricFigure):
  def __init__(self, …):
```

Classes in C++

- C++ is object-based, *classes* can be implemented
- *classes* have *declaration* and *definition*
- *classes* have the same components as in Python:

  data members = *attributes* = (*instance* or *class*) *variables*
  member functions = (*instance* or *class*) *methods*
- a *class declaration* is usually written in a *header file* (<classname>.h). It declares *data members* and *member functions*.
- The definitions of *member functions* of a *class* are usually in an *implementation file* (<classname>.cpp).

Classes in C++

- *access levels* (*public, private, protected*) are set for *attributes* and *member functions*
- const member functions do not change any data member (attribute) of the object.
- no self parameter in member functions

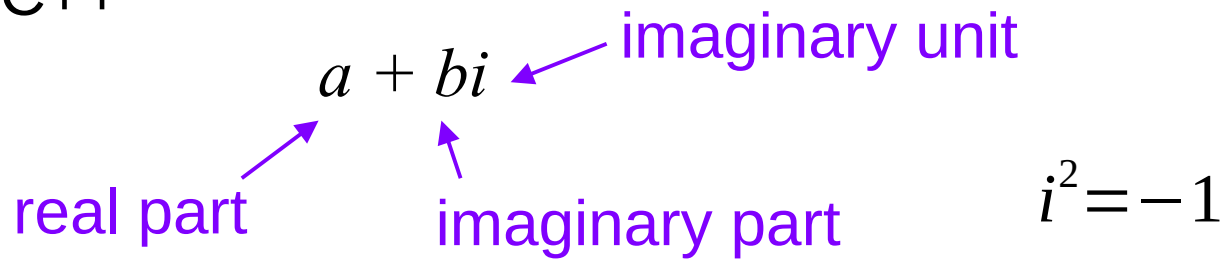
Classes in C++

```
class Dice: public GeometricFigure
{
  Dice(…) { … }
};
```

with public base class, public members of the base class become public members of the derived class; protected members of the base class become protected members of the derived class.
A base class's private members are never accessible directly from a derived class, but can be accessed through calls to the public and protected members of the base class.

Complex number

- Let’s implement Complex number in its rectangular form in both, Python and in C++



Complex number

$a + bi$

- Let’s implement Complex number in its rectangular form in both, Python and in C++

To-do list:

Complex number

$a + bi$

- Let's implement Complex number in its rectangular form in both, Python and in C++

To-do list:

- declare/define two attributes (`private`): real part, imaginary part
- declare/define initializer/constructor that enables an object of this class to be *initialized when it's declared*; provide *default values* in case no initializers are provided
- provide (`public`) member functions that perform the following tasks:
  - `add`: adds two complex numbers,
  - `subtract`: subtracts the two complex numbers
  - `multiply`: multiplies the two complex numbers
  - `divide`: divides the two complex numbers, ...
- We should be able to display/print the instance of the Complex class
- We should be able to get a copy of the instance of the Complex class

Complex number

$a + bi$

- multiplication and division of complex numbers?

(don't forget that division by 0 is undefined)

$$(a+bi)(c+di)=\ldots=(ac-bd)+(ad+bc)i$$

$$\frac{a+bi}{c+di}=\frac{(a+bi)(c-di)}{(c+di)(c-di)}=\ldots=\frac{(ac+bd)+(bc-ad)i}{c^2+d^2}=$$

$$=\frac{ac+bd}{c^2+d^2}+\frac{bc-ad}{c^2+d^2}i$$