



FULL SAIL
UNIVERSITY

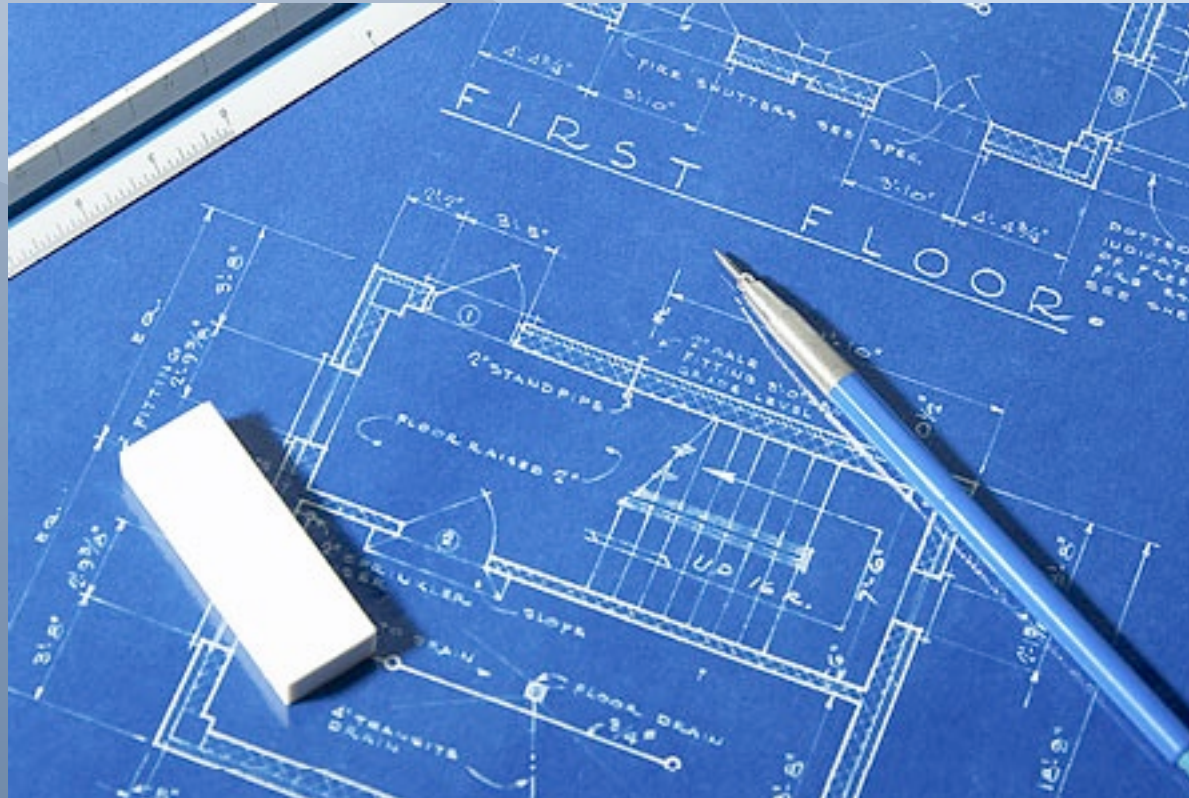
Classes & Class Members

Creating templates for objects

Design Patterns for Web Programming
Web Design & Development Bachelor of Science Degree

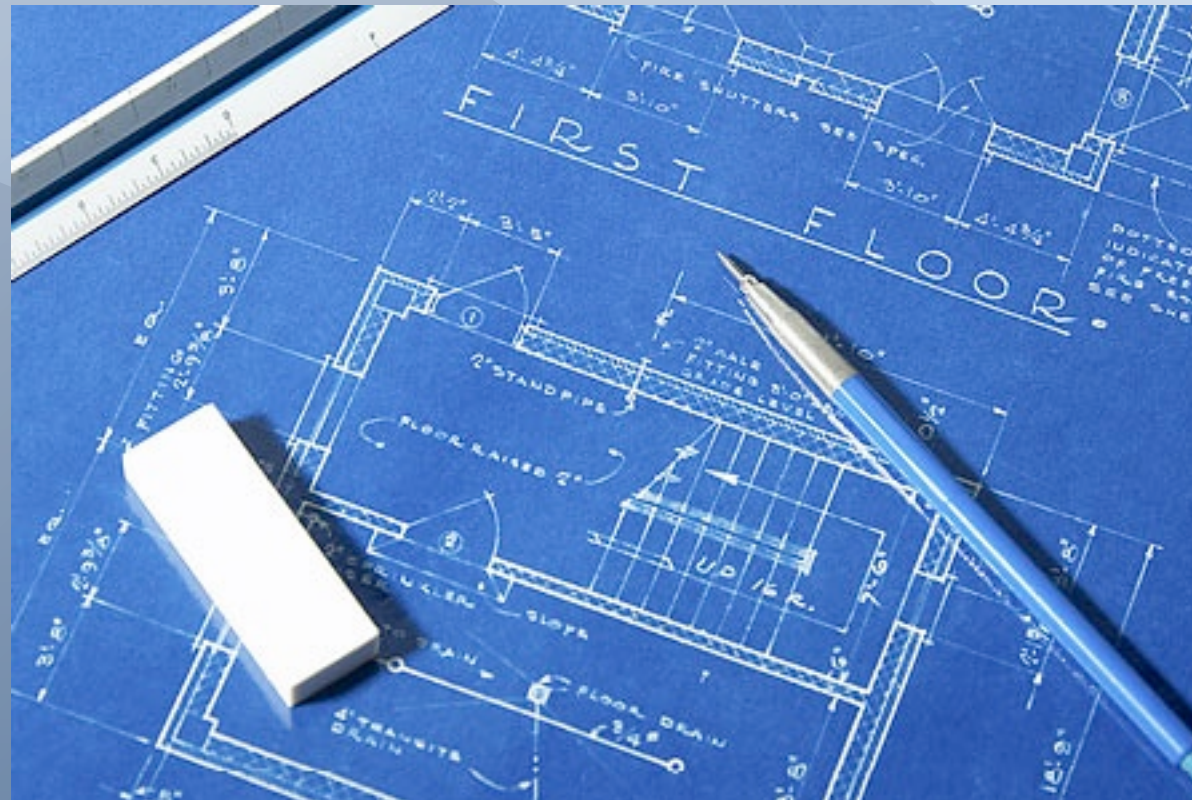


Classes



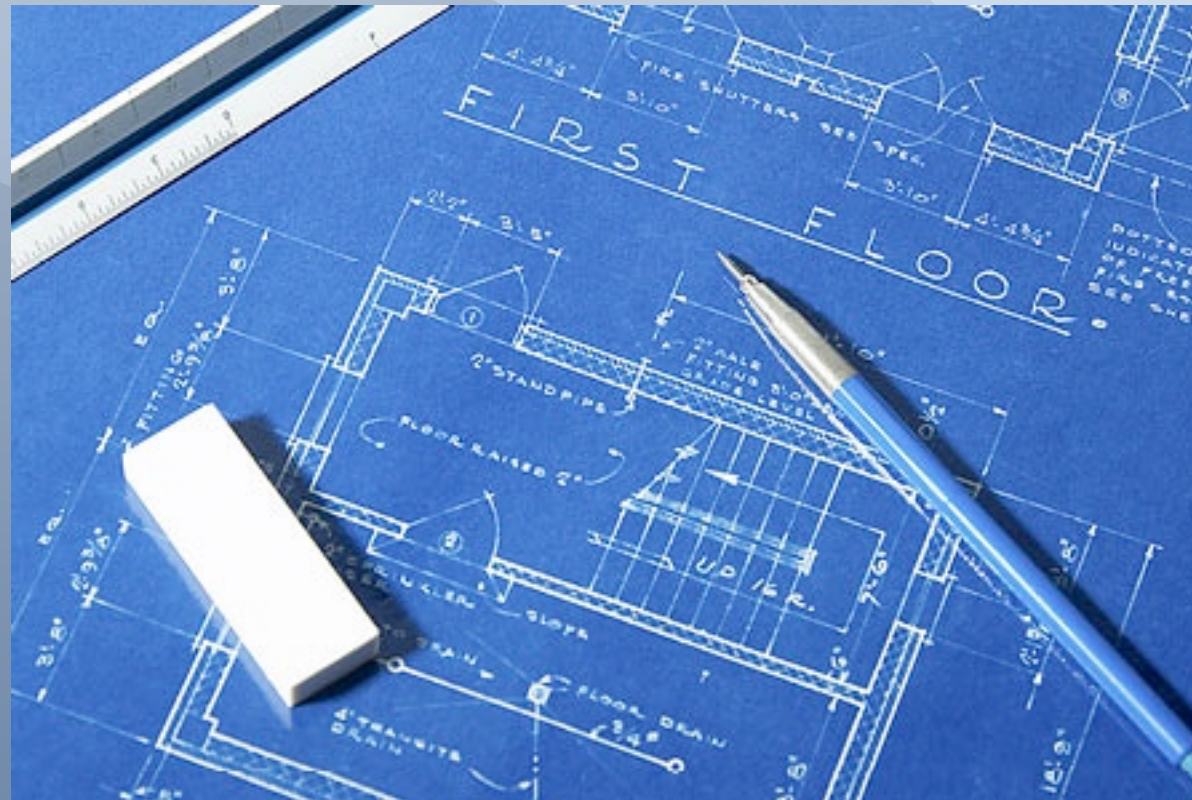
Classes

- A Class is a set of instructions for building an object, what traits the object has and what behaviors the object can *do*.



Classes

- A Class is a set of instructions for building an object, what traits the object has and what behaviors the object can *do*.



Instances



Instances

- Many ***instances*** can be created from a class.



Instances

- Many **instances** can be created from a class.
 - ✓ For Example: you might have a `contact_button` and an `about_button`, but they are both instances of a the class `MyButton`.



Instances

- Many **instances** can be created from a class.
 - ✓ For Example: you might have a `contact_button` and an `about_button`, but they are both instances of a the class `MyButton`.

```
contact_button = Button()
```



Instances

- Many **instances** can be created from a class.
 - ✓ For Example: you might have a `contact_button` and an `about_button`, but they are both instances of a the class `MyButton`.

```
contact_button = Button()
```

```
about_button = Button()
```





Class Members

Flash Fundamentals
Web Design & Development Bachelor of Science Degree



Class Members

- An object will expose information and functionality pertaining to that object in the form of:



Class Members

- An object will expose information and functionality pertaining to that object in the form of:
 - Attributes



Class Members

- An object will expose information and functionality pertaining to that object in the form of:
 - Attributes
 - Properties (we will discuss in the Encapsulation lecture)



Class Members

- An object will expose information and functionality pertaining to that object in the form of:
 - Attributes
 - Properties (we will discuss in the Encapsulation lecture)
 - Methods



Attributes

- Attributes of an object are values that describe the characteristics of that object.
- Traits
- Variables
- Dot Syntax:
 - `object.attribute = value;`
- Attributes are objects, too, and can have properties of their own.
 - `object.attribute.attribute = value;`
 - `object.attribute.attribute += 1;`



Access to Modifiers

Who can get to what?

○ Public

- accessible by any other object
- **`self.attribute`**

○ Private

- accessible ONLY within the class it is defined
- **`self.__attribute` (two underscores!!)**

○ Protected

- accessible by sub-classes.
- **`self._attribute`**



Methods



Methods

- A method is a named block of code that can be called from other places in your program.



Methods

- A method is a named block of code that can be called from other places in your program.
- Behaviors



Methods

- A method is a named block of code that can be called from other places in your program.
- Behaviors
- Functions



Methods

- A method is a named block of code that can be called from other places in your program.
- Behaviors
- Functions
- Methods can accept input (called parameters or arguments) and provide output (by returning a value)



Methods

- A method is a named block of code that can be called from other places in your program.
- Behaviors
- Functions
- Methods can accept input (called parameters or arguments) and provide output (by returning a value)
- Dot syntax:



Methods

- A method is a named block of code that can be called from other places in your program.
- Behaviors
- Functions
- Methods can accept input (called parameters or arguments) and provide output (by returning a value)
- Dot syntax:
 - `object.method(value);`



Methods

- A method is a named block of code that can be called from other places in your program.
- Behaviors
- Functions
- Methods can accept input (called parameters or arguments) and provide output (by returning a value)
- Dot syntax:
 - `object.method(value);`
 - `object.method();`



Methods

- A method is a named block of code that can be called from other places in your program.
- Behaviors
- Functions
- Methods can accept input (called parameters or arguments) and provide output (by returning a value)
- Dot syntax:
 - `object.method(value);`
 - `object.method();`
 - `object.attribute.method();`





Constructor

Flash Fundamentals
Web Design & Development Bachelor of Science Degree





Constructor

○ Catalyst

Flash Fundamentals
Web Design & Development Bachelor of Science Degree



Constructor

- Catalyst
 - When an object is created this method runs immediately.



Constructor

- Catalyst
 - When an object is created this method runs immediately.
- Holds attributes available to the entire class.



Constructor

- Catalyst
 - When an object is created this method runs immediately.
- Holds attributes available to the entire class.
- **def `__init__`(self):**





Class Members

Attributes & Methods

Flash Fundamentals
Web Design & Development Bachelor of Science Degree



Class Members

Attributes & Methods

```
class Main(object):  
    def __init__(self):  
        self.greeting = "Hello";  
        self.changeGreetingToFrench();  
  
    def changeGreetingToFrench(self):  
        self.greeting = "Bonjour";
```





Class Members

Properties

Flash Fundamentals
Web Design & Development Bachelor of Science Degree



Class Members

Properties

```
class Main(object):  
    def __init__(self):  
  
        self.greeting = "Hello";  
        self.changeGreetingToFrench();  
  
    def changeGreetingToFrench(self):  
        self.greeting = "Bonjour";
```



Class Members

Properties

```
class Main(object):  
    def __init__(self):  
        self.greeting = "Hello";  
        self.changeGreetingToFrench();  
  
    def changeGreetingToFrench(self):  
        self.greeting = "Bonjour";
```





Class Members

Methods

Flash Fundamentals
Web Design & Development Bachelor of Science Degree



Class Members

Methods

```
class Main(object):  
  
    def __init__(self):  
        self.greeting = "Hello";  
        self.changeGreetingToFrench();  
  
    def changeGreetingToFrench(self):  
        self.greeting = "Bonjour";
```



Class Members

Methods

```
class Main(object):
```

```
    def __init__(self):  
        self.greeting = "Hello";  
        self.changeGreetingToFrench();
```

```
    def changeGreetingToFrench(self):  
        self.greeting = "Bonjour";
```



Class Members

Methods

```
class Main(object):
```

```
    def __init__(self):  
        self.greeting = "Hello";  
        self.changeGreetingToFrench();
```

```
    def changeGreetingToFrench(self):  
        self.greeting = "Bonjour";
```





Review

Flash Fundamentals
Web Design & Development Bachelor of Science Degree



Review

- Classes are blueprints for instances



Review

- Classes are blueprints for instances
- Attributes are special variables for traits



Review

- Classes are blueprints for instances
- Attributes are special variables for traits
- Methods are special functions for behaviors



Review

- Classes are blueprints for instances
- Attributes are special variables for traits
- Methods are special functions for behaviors
- Constructor methods are for setting up an object

