

CREATE CUSTOM PYTHON CLASS CHEAT SHEET

1. Create a normal Python module file.
2. Define any “CONSTANTS” used in classes or unit tests.
3. Define the class: `class ClassName:`
4. Define the constructor: `def __init__(self, initval1, initval2, ...)`
 - a. Set value of each instance variable in the constructor.
 - b. Use `int()`, `float()`, `str()`, etc. to control value types of instance variables.
5. Define the `str()` method by creating the `__str__()` method.
6. Define the `repr()` method by creating the `__repr__()` method.
7. Define a calculate method for each property of the class that is calculated rather than stored (eg. `calculate_total_weight()`).
8. Consider adding any further methods that provide convenience for the class user or that implement advanced Python class features taught in more advanced Python courses.
9. If (and only if) your class requires special getter/setter code for instance variables, then add **implicit getter/setter code** for only those instance variables for which it is required using the **@property** notation.
10. Place all of your unit testing code in the `main()` method. This is the convention that we use in our course. Professional programmers place this in separate unit testing framework classes.
11. Invoke `main()` method with the final statements in your module making sure that unit test code only runs when the module is run by itself and not when the module is imported by another program.

```
if __name__ == '__main__':  
    main()
```

Unit Testing Hints

1. Every time that you add a method to the class code, immediately build and run the code for related unit test cases.
2. Simple methods may need only one unit test case. Others methods may need more.
3. Remember to create a separate test case for every error condition checked by your code.