Base Overloading Methods

Following table lists some generic functionality that you can override in your own classes:

Sr. No.	**Method, Description, and Sample Call**
1	__init__ (self [,args...]) Constructor (with any optional arguments) Sample Call : *obj = className(args)*
2	__del__(self) Destructor, deletes an object Sample Call : *del obj*
3	__repr__(self) Evaluatable string representation Sample Call : *repr(obj)*
4	__str__(self) Printable string representation Sample Call : *str(obj)*
5	__cmp__ (self, x) Object comparison Sample Call : *cmp(obj, x)*

Overloading Operators

Suppose you have created a Vector class to represent two-dimensional vectors, what happens when you use the plus operator to add them? Most likely Python will yell at you.

You could, however, define the *__add__* method in your class to perform vector addition and then the plus operator would behave as per expectation:

Example

```
#!/usr/bin/python



class Vector:
```