

## Recipe for A New Object-Oriented Class (a new Type)

1. Create the shell of the class using Eclipse. Remember that there is no *main* method in an object-oriented class.
2. Add the instance variables to the top of your code – just below the class header. These are the fields that hold the values for each instance. They should include the *private* modifier.
3. If there are any fields (such as constants) that are to be visible to all instances of a class, add them now. Remember that fields visible to all instances must include the modifier *static*. Also, `CONSTANT_FIELDS` should be both *static* and *final*. You can add these fields either just before or just after the instance variables depending upon your taste.
4. Add getters and setters for each instance variable. Use the Eclipse wizard to do this.
5. Add any constructors that you will need. Place them after the instance variables and before the getters and setters. This may seem out of sequence. But, it is the traditional order in which constructors appear. Constructors that you will need include:
  - a. A No-Argument Constructor - While this is technically not required when it will be the only constructor, it is a good practice to always include an explicit No-Argument Constructor.
  - b. Constructors with Arguments – These allow client code to initialize instance variables at the same that the new object is constructed. The number of these constructors needed (if any) will depend on the requirements for the class.
6. Add a *toString()* method. Remember that this overrides the *toString()* method in Object class. Use Eclipse to generate your *toString()*. Note that this generated method has an `@Override` annotation above it.
7. Create a testing client class and test the code that you have created so far.
8. If you need to implement an *equals()* method, then use Eclipse to Generate both *hashCode()* and *equals()*. After doing so, test the *equals()* method.
9. Add any additional methods (worker methods) that are required. Place them at the bottom of the class after the *equals()* method. After doing so, test the new methods.