Object Oriented Programming in PHP - Copying and Cloning

When you create an object $obj you can copy the object by doing $obj2=$obj, the new object is a copy (not a reference) of $obj so it has the state $obj had in the moment the assignment was made. Sometimes you don't want this you just want to create a new object of the same class as obj, calling the constructor of the new object as if you had used the new statement.
This can be done in PHP using serialization and a base class that all other classes must extend.

Entering a Danger Zone

When you serialize an object you get a string which has a certain format, you may investigate this if you are curious. One of the things the string has is the name of the class (nice!), you can extract it using:

```
<?php

$herring=serialize($obj);
$vec=explode(':',$herring);
$nam=str_replace("\"","",$vec[2]);

?>
```

So suppose you create a class "Universe" and force that all classes must extend universe, you can define a method clone in Universe as:

```
<?php

class Universe {
function clone() {
$herring=serialize($this);
$vec=explode(':',$herring);
$nam=str_replace("\"","",$vec[2]);
$ret=new $nam;
return $ret;
}
}
```

Then:

```
$obj=new Something();
//Something extends Universe !!
$other=$obj->clone();

?>
```

What you get is a new object of class Something created the same way as using new; the constructor is called, etc. I don't know if this is useful for you but the Universe class which knows the name of the derived class is a nice concept to experiment with. The only limit is your imagination.

Note: I'm using PHP4, some of these examples may not work in PHP3.