```php
<?php
// ABSTRACTION
abstract class Appliance {
  protected $brand;
  protected $power;  // in watts
  // ENCAPSULATION
  public function __construct($brand, $power) {
     $this->setBrand($brand);
     $this->setPower($power);
  }
  // ENCAPSULATION
  public function getBrand() {
     return $this->brand;
  }
  public function setBrand($brand) {
     if (!empty($brand)) {
        $this->brand = $brand;
     }
  }
  public function getPower() {
     return $this->power;
  }
  public function setPower($power) {
     if ($power > 0) {
        $this->power = $power;
     }
  }
  // Abstract method to be implemented by subclasses (Abstraction)
  abstract public function operate();
}
// INHERITANCE
class WashingMachine extends Appliance {
  public function operate() {
     return $this->brand . " washing machine is running with power " . $this->power . "
watts.";
  }
}
class Refrigerator extends Appliance {
  public function operate() {
     return $this->brand . " refrigerator is cooling with power " . $this->power . " watts.";
  }
}
// POLYMORPHISM
```

```
function testAppliance(Appliance $appliance) {
  echo $appliance->operate() . "\n";
}
// Create objects
$washer = new WashingMachine("LG", 1500);
$fridge = new Refrigerator("Samsung", 800);
testAppliance($washer);  // Output: LG washing machine is running with power 1500 watts.
testAppliance($fridge);  // Output: Samsung refrigerator is cooling with power 800 watts.
?>
```