

IS113 - Web Application Development

Week 9

Sections G4/G5

K. J. Shim

Announcements

- **Quiz 3** next week (**Week 10**)
 - Scope: **Week 9 lesson** (Classes & Objects)
- **Lab Test 1**
 - The following will be released to all students by **16-Mar-2021 (Tuesday) 5PM SG Time**
 1. Solutions
 2. Component marks (Q1/Q2/Q3)
 3. Grading rubrics
 4. Common mistakes

Agenda Today

- Classes & Objects
 - Basic concepts
 - Exercises

Go to your **webroot**

- Go to eLearn -> Content -> Session 9 -> **week9.zip**
- Unzip **week9.zip** such that you have the following directory structure locally:

```
webroot -  
|-- is113  
|   |-- week9  
|       |-- mama  
|           |-- display.php  
|           |-- process.php  
|           |-- data.php  
|           |-- Item.php  
|           |-- test.php
```

Exercise 1

mama

So far...

display.php

```
$fruits = [ 'apple', 'orange', 'pear' ];  
...  
...
```

```
if( isset($_POST['age']) ) {  
    $age = $_POST['age'];  
    ...  
}
```

```
<html>  
...  
  
<table border='1'>  
    <tr>  
        <td> ...  
    </td>  
</table>  
...
```

Store Data

e.g. Arrays

Form Processing

e.g. form validation

Display Data

e.g. Lists, Table, Images...

Going forward...

data.php

```
$fruits = [ 'apple', 'orange', 'pear' ];  
...  
...
```

Store Data

e.g. Arrays

display.php

```
if( isset($_POST['age']) {  
    $age = $_POST['age'];  
    ...  
}
```

Form Processing

e.g. form validation

```
<html>  
...  
<table border='1'>  
    <tr>  
        <td> ...  
    </td>  
</table>  
...
```

Display Data

e.g. Lists, Table, Images...

Solution #1

data.php

- Store **all items' data** in a **multi-dimensional Array**
- Makes the **Array** available to whoever that needs to display **items' data**.

requires



display.php

- List all **items** in an HTML table
- Does NOT contain any data
- It simply **displays** items

- Let's separate item **data** from **display.php** and put it into **data.php**

data.php

```
$itemsArr = [  
    'A001' => [  
        'description' => "Supreme Shampoo",  
        'price' => 5.75,  
        'inventory' => 12  
    ],  
    'B023' => [  
        'description' => "Supreme Toothbrush",  
        'price' => 3.50,  
        'inventory' => 5  
    ],  
    'C456' => [  
        'description' => "Supreme Pencil",  
        'price' => 1.25,  
        'inventory' => 7  
    ]  
];
```

- Complete **display.php** such that it pulls **item data** from **\$itemsArr** defined in **data.php**

data.php

```
$itemsArr = [  
    'A001' => [  
        'description' => "Supreme Shampoo",  
        'price' => 5.75,  
        'inventory' => 12  
    ],  
    'B023' => [  
        'description' => "Supreme Toothbrush",  
        'price' => 3.50,  
        'inventory' => 5  
    ],  
    'C456' => [  
        'description' => "Supreme Pencil",  
        'price' => 1.25,  
        'inventory' => 7  
    ]  
];
```

- When completed, **display.php** looks like:

Pyongyang's Best Mama Shop

Items Available

Item ID	Item Description
A001	<input type="checkbox"/> Supreme Shampoo
B023	<input type="checkbox"/> Supreme Toothbrush
C456	<input type="checkbox"/> Supreme Pencil

Your Particulars

☐ Local ☐ Foreigner

Your Name:

The form submits to **process.php**.

The user must:

- Select at least 1 **item** (checkbox)
- Select residency status (**local** or **foreigner**)
- Input a **name** (non-empty)

Your Particulars

☐ Local ☐ Foreigner

```
<label>
|   <input type="radio" name="residency" value="local"> Local
</label>

<label>
|   <input type="radio" name="residency" value="foreigner"> Foreigner
</label>
```

- Instead of clicking on the **radio circle button**, try clicking on the **text**, e.g. **Local** or **Foreigner**
- You can link a form input field with a **label**.

display.php

Items Available

Item ID	Item Description
A001	<input type="checkbox"/> Supreme Shampoo
B023	<input type="checkbox"/> Supreme Toothbrush
C456	<input type="checkbox"/> Supreme Pencil

Your Particulars

☐ Local ☐ Foreigner

Your Name:

Your Name field is left **EMPTY**
(not even whitespaces)

process.php

1. You need to select at least 1 item
2. You need to indicate residency status
3. Your name cannot be empty

display.php

Items Available

Item ID	Item Description
A001	<input checked="" type="checkbox"/> Supreme Shampoo
B023	<input checked="" type="checkbox"/> Supreme Toothbrush
C456	<input type="checkbox"/> Supreme Pencil

Your Particulars

☐ Local ☐ Foreigner

Your Name:

Your Name field is left **EMPTY**
(not even whitespaces)

process.php

1. You need to indicate residency status
2. Your name cannot be empty

display.php

process.php

Items Available

Item ID	Item Description
A001	<input checked="" type="checkbox"/> Supreme Shampoo
B023	<input type="checkbox"/> Supreme Toothbrush
C456	<input checked="" type="checkbox"/> Supreme Pencil

Your Particulars

☒ Local ☐ Foreigner

Your Name:

Your Name field has **several whitespaces**.

1. Your name cannot be empty

display.php

Items Available

Item ID	Item Description
A001	<input checked="" type="checkbox"/> Supreme Shampoo
B023	<input type="checkbox"/> Supreme Toothbrush
C456	<input checked="" type="checkbox"/> Supreme Pencil

Your Particulars

☒ Local ☐ Foreigner

Your Name:

process.php

Hello Bae Soo Ji!

Current Inventory for the selected items are:

Item ID	Item Description	Price (USD)	Current Inventory
A001	Supreme Shampoo	5.75	12
C456	Supreme Pencil	1.25	7

Your tax rate is 5%

- Greet the user with his/her **name**
- For the selected **items**, display **price** and **current inventory** data.
- Tax rate is **5%** for **local** and **30%** for **foreigner**.

display.php

Items Available

Item ID	Item Description
A001	<input type="checkbox"/> Supreme Shampoo
B023	<input checked="" type="checkbox"/> Supreme Toothbrush
C456	<input checked="" type="checkbox"/> Supreme Pencil

Your Particulars

☐ Local ☒ Foreigner

Your Name:

process.php

Hello Kaneshiro Takeshi!

Current Inventory for the selected items are:

Item ID	Item Description	Price (USD)	Current Inventory
B023	Supreme Toothbrush	3.5	5
C456	Supreme Pencil	1.25	7

Your tax rate is 30%

- Greet the user with his/her **name**
- For the selected **items**, display **price** and **current inventory** data.
- Tax rate is **5%** for **local** and **30%** for **foreigner**.

Solution #2

Item.php

- Provides **definition** of what an **item** should be
- This file alone does NOT give us any actual **items**

requires

data.php

- Borrows the **item definition** from Item.php
- Creates **new items** (“Item objects”)
- Stores the **new items** in a **data structure** (e.g. Array)
- Makes the **Array** available to whoever that needs to display **items**

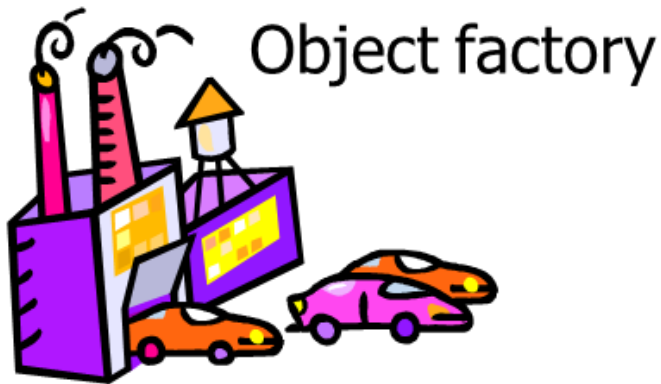
requires

display.php

- List all **items** in an HTML table
- Does NOT contain any data
- It simply **displays** items

Classes

- A class is a **template** from which individual objects are created
- An object is an **instance** of a class



Cookie Cutter



What is an Object?

- Informally, an object represents an entity, either physical, conceptual, or software.

- Physical entity



Tom's Car

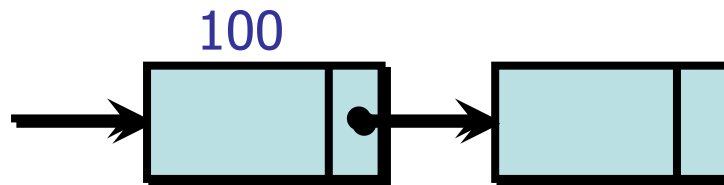
- Conceptual entity



US\$100,000,000,000

Bill Gate's bank account

- Software entity



Linked list starting at address 100

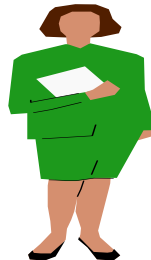
What is an Object?

- All the objects share the same **properties** and **methods** with other objects of the same class
- Each object has its own value for each of the attribute

Person
- name - dateOfBirth
+ greet()



harry:Person	
name	Harry
dateOfBirth	8/12/1975
+ greet()	



mary:Person	
name	Mary
dateOfBirth	8/12/1980
+ greet()	

<u>myMan:Man</u>	
name	Hyun Bin
wife_name	Ye Jin

← \$myMan

<u>yourMan:Man</u>	
name	Justin
wife_name	Selena

← \$yourMan

Class Definition + Properties and Methods

- Person.php -

```
<?php
```

```
class Person {
```

```
    private $name;
```

```
    private $dateOfBirth;
```

```
    public function greet(){
```

```
        echo "Hi! My name is $this->name.";
```

```
    }
```

```
    public function getDateOfBirth() {
```

```
        return $this->dateOfBirth;
```

```
    }
```

```
    ...
```

```
}
```

Properties



Methods



```
?>
```

Constructor – special method

- Constructor

- Method that creates an instance of a class (an object)
- Can accept parameters to initialize class properties

- Person.php -

```
<?php
class Person {

    private $name;
    private $dateOfBirth;

    public function __construct($name, $dateOfBirth) {
        $this->name = $name;
        $this->dateOfBirth = $dateOfBirth;
    }
}
?>
```


Creating new **Person** objects

- **calling_method.php** -

```
<?php
```

```
require_once "Person.php";
```

```
$harry = ...
```

```
?>
```

Creating new **Person** objects

- calling_method.php -

```
<?php
```

```
require_once "Person.php";
```

```
$harry = new Person(  
    'Harry',  
    '8/12/1975');  
  
?>
```

<u>harry:Person</u>	
name	<input type="text"/>
dateOfBirth	<input type="text"/>
+ greet()	

Creating new **Person** objects

- calling_method.php -

```
<?php  
require_once "Person.php";
```

```
$harry = new Person(  
    'Harry',  
    '8/12/1975');
```

<u>harry:Person</u>	
name	<input type="text"/>
dateOfBirth	<input type="text"/>
+ greet()	

- Person.php -

```
<?php  
class Person {  
    ...  
    public function __construct($name, $dateOfBirth) {  
        $this->name = $name;  
        $this->dateOfBirth = $dateOfBirth;  
    }  
    ...  
}
```

?>

Creating new **Person** objects

- calling_method.php -

```
<?php  
require_once "Person.php";
```

```
$harry = new Person(  
    'Harry',  
    '8/12/1975');
```

<u>harry:Person</u>	
name	<input type="text"/>
dateOfBirth	<input type="text"/>
+ greet()	

- Person.php -

```
<?php  
class Person {  
    ...  
    public function __construct($name, $dateOfBirth) {  
        $this->name = $name;  
        $this->dateOfBirth = $dateOfBirth;  
    }  
    ...  
}
```

?>

Creating new Person objects

- calling_method.php -

```
<?php  
require_once "Person.php";
```

```
$harry = new Person(  
    'Harry',  
    '8/12/1975');
```

harry:Person	
name	<input type="text"/>
dateOfBirth	<input type="text"/>
+ greet()	

- Person.php -

```
<?php  
class Person {  
    ...  
    public function __construct($name, $dateOfBirth) {  
        $this->name = $name;  
        $this->dateOfBirth = $dateOfBirth;  
    }  
    ...  
}
```

?>

Creating new Person objects

- calling_method.php -

```
<?php  
require_once "Person.php";
```

```
$harry = new Person(  
    'Harry',  
    '8/12/1975');
```

<u>harry:Person</u>	
name	<input type="text"/>
dateOfBirth	<input type="text"/>
+ greet()	

- Person.php -

```
<?php  
class Person {  
    ...  
    public function __construct($name, $dateOfBirth) {  
        $this->name = $name;  
        $this->dateOfBirth = $dateOfBirth;  
    }  
    ...  
}
```

?>

Creating new Person objects

- calling_method.php -

```
<?php  
require_once "Person.php";
```

```
$harry = new Person(  
    'Harry',  
    '8/12/1975');
```

harry:Person	
name	Harry
dateOfBirth	8/12/1975
+ greet()	

- Person.php -

```
<?php  
class Person {  
    ...  
    public function __construct($name, $dateOfBirth) {  
        $this->name = $name;  
        $this->dateOfBirth = $dateOfBirth;  
    }  
    ...  
}
```

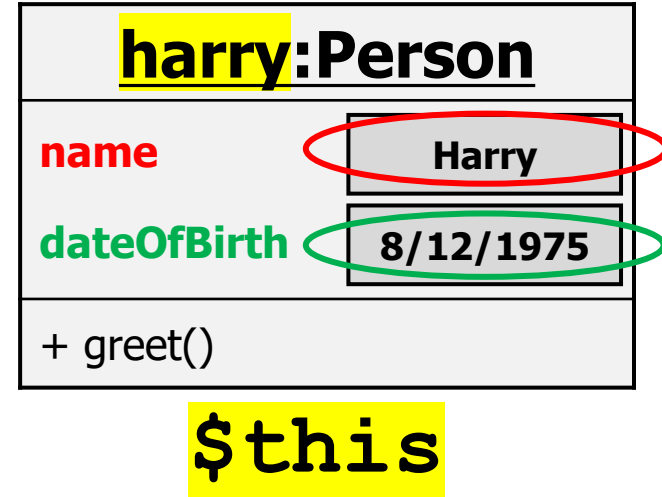
?>

Creating new Person objects

- calling_method.php -

```
<?php  
require_once "Person.php";
```

```
$harry = new Person(  
    'Harry',  
    '8/12/1975');
```



- Person.php -

```
<?php  
class Person {  
    ...  
    public function __construct($name, $dateOfBirth) {  
        $this->name = $name;  
        $this->dateOfBirth = $dateOfBirth;  
    }  
    ...  
}
```

?>

Creating new Person objects

- calling_method.php -

```
<?php  
require_once "Person.php";
```

```
$mary = new Person(  
    'Mary',  
    '8/12/1980');  
?  

```

mary:Person	
name	Mary
dateOfBirth	8/12/1980
+ greet()	

\$this

- Person.php -

```
<?php  
class Person {  
    ...  
    public function __construct($name, $dateOfBirth) {  
        $this->name = $name;  
        $this->dateOfBirth = $dateOfBirth;  
    }  
    ...  
}
```

Access Modifier: Private vs. Public

- Man.php -

```
<?php
```

```
class Man {
```

```
    private $name;
```

```
    public $wife_name;
```

```
    public function getName() {  
        return $this->name;  
    }
```

```
}
```

```
}
```

```
?>
```

Access
Modifier

Access Modifier: Private vs. Public

- Creator.php -

```
<?php
    $man = new Man('Hyun Bin', 'Ye Jin');
?>
```

- Stranger.php -

```
<?php
    require_once 'Creator.php';

    // What's your name?
    echo $man->getName();

    // What's your wife's name?
    echo $man->wife_name;

?>
```

Access Modifier: Private vs. Public

- Stranger.php -

```
<?php
    require_once 'Creator.php';

    // What's your wife's name?
    echo $man->wife_name;

    // Hehehe... Let me replace your wife...
    $man->wife_name = 'Kyong';

    // Now... Who is your wife? Huh?
    echo $man->wife_name;

?>
```

Access Modifier: Private vs. Public

- Man.php -

```
<?php
```

```
class Man {
```

```
    private $name;
```

```
    private $wife_name;
```

```
    public function getName() {  
        return $this->name;  
    }
```

```
    public function getWifeName() {  
        return $this->wife_name;  
    }
```

```
}
```

```
?>
```

Access
Modifier

Access Modifier: Private vs. Public

Now... try changing \$man's wife_name ... Does it work?

- Stranger.php -

```
<?php
    require_once 'Creator.php';

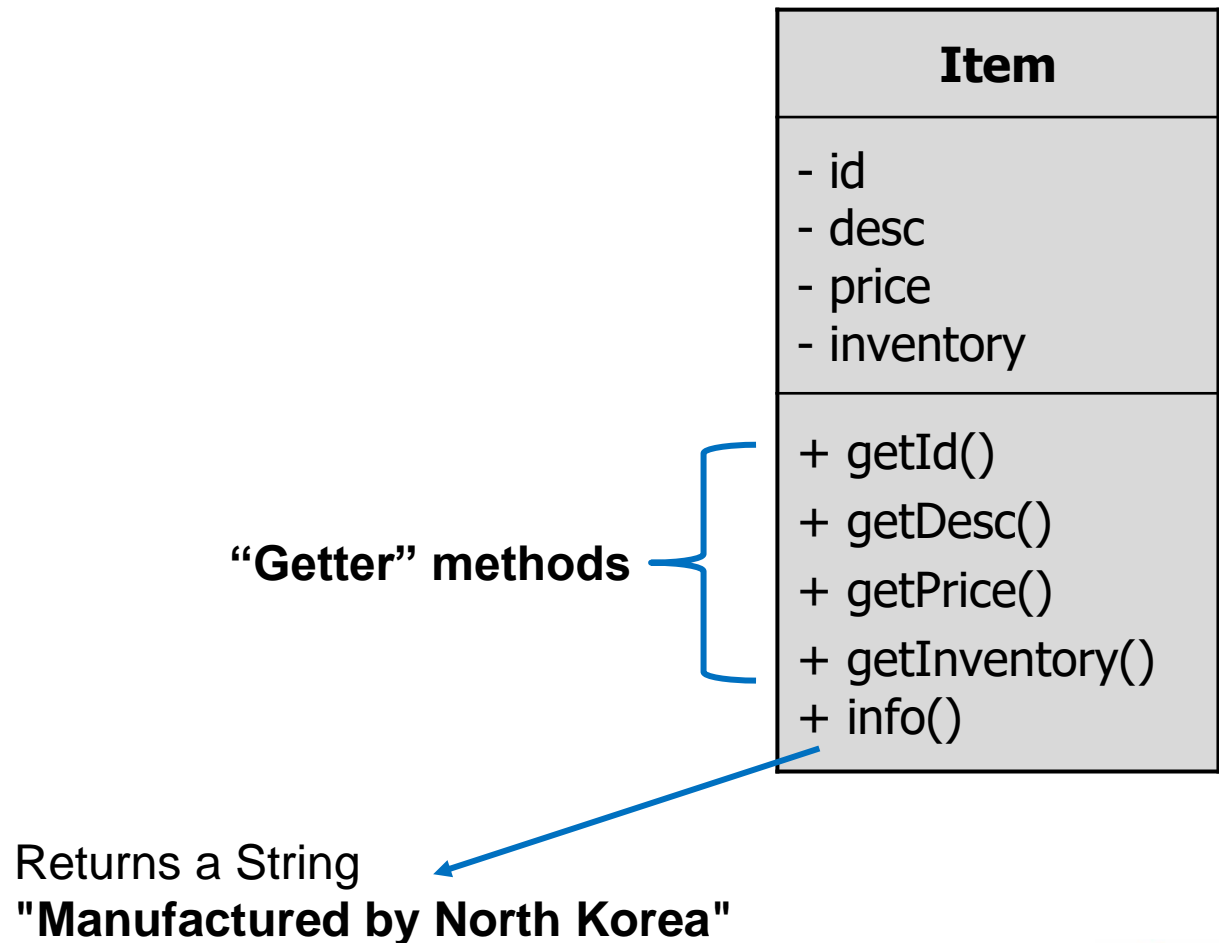
    // What's your wife's name? ERROR
    echo $man->getWifeName();

    // Hehehe... Let me replace your wife...
    $man->wife_name = 'Kyong';

    // Now... Who is your wife? Huh?
    echo $man->getWifeName();

?>
```

- Complete the definition of **Item Class** according to the below diagram:



- Create two **new Item objects** with the following information:

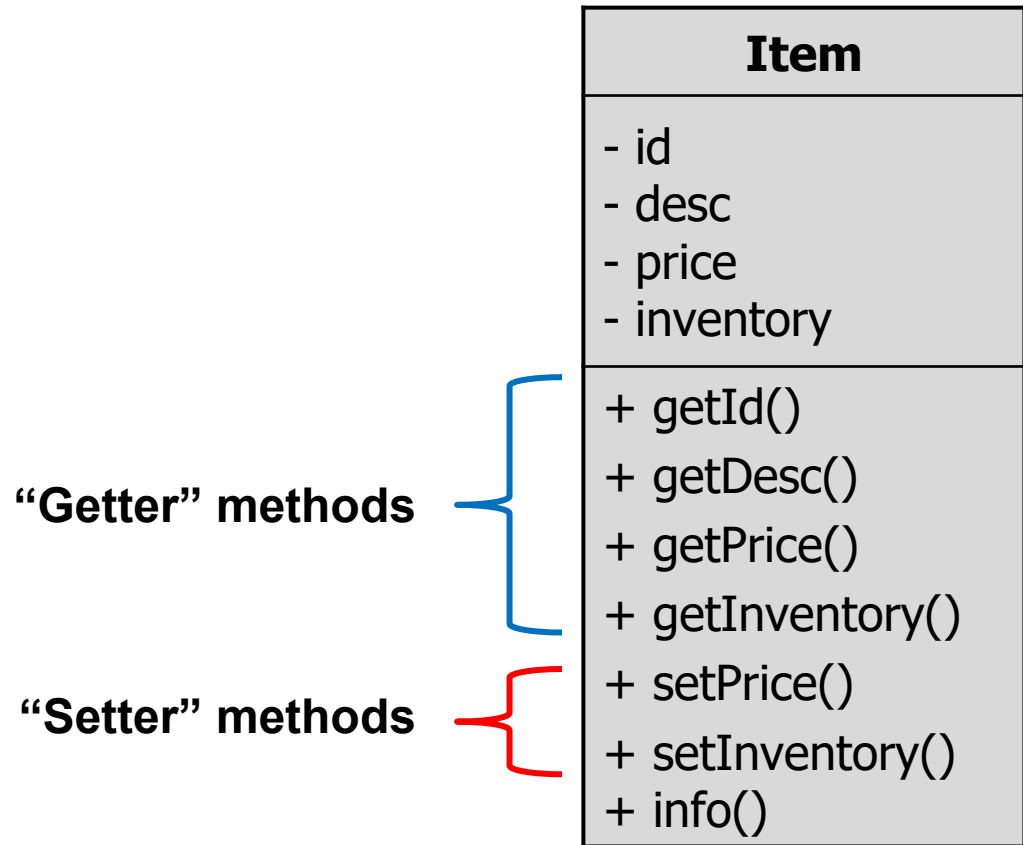
<u>myItem:Item</u>	
id	X123
desc	Supreme Coffee
price	15.75
inventory	30
+ info()	

<u>yourItem:Item</u>	
id	Y456
desc	Supreme Knife
price	22.50
inventory	20
+ info()	

- How can we retrieve and print **myItem's desc** and **price**?

```
require_once 'Item.php';
$myItem = new Item('X123', 'Supreme Coffee', 15.75, 30);
$yourItem = new Item('Y456', 'Supreme Knife', 22.50, 20);
echo $myItem->getDesc() . " costs USD " . $myItem->getPrice() . "<br>;
```


- Modify **Class Item** such that it contains **Setter methods** for **price** and **inventory**.



- Update the **price** of **myItem** to **20.99** and **inventory** to **10**

<u>myItem:Item</u>	
id	X123
desc	Supreme Coffee
price	20.99
inventory	10
+ info()	

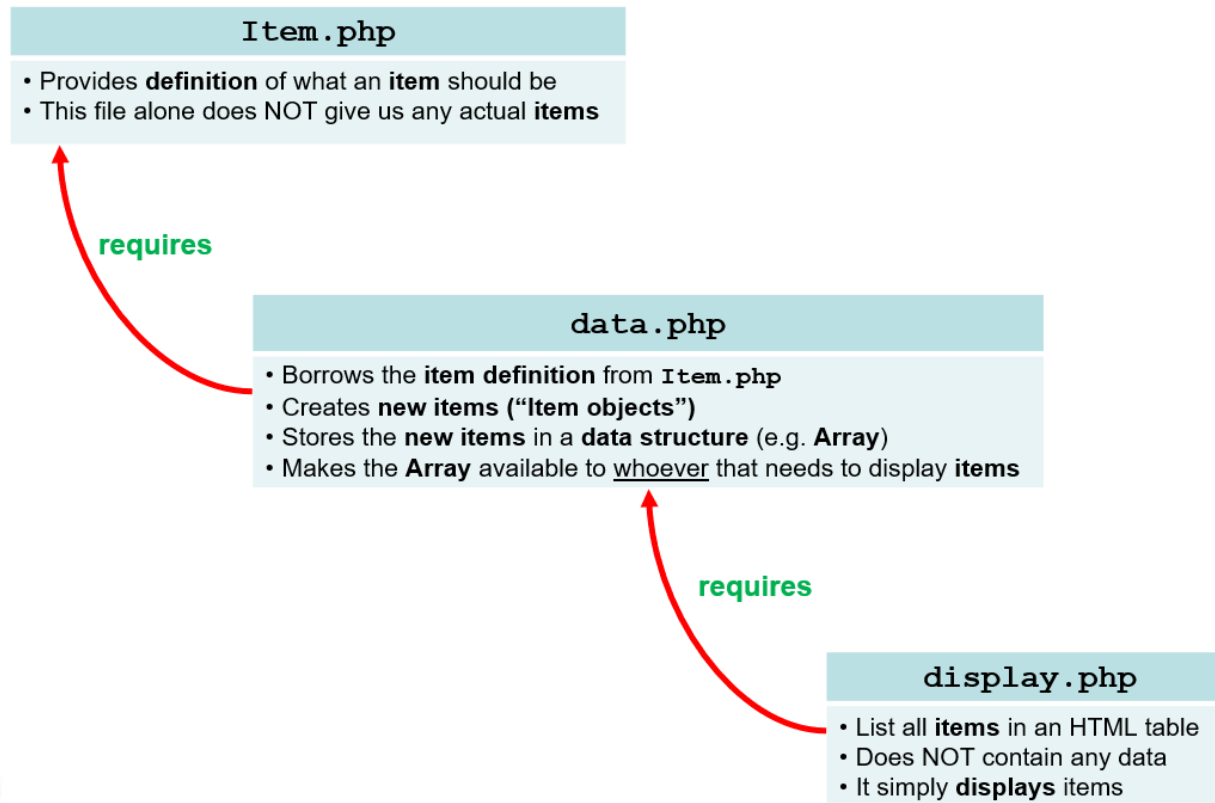
```
echo "Before update, myItem price USD: " . $myItem->getPrice() . "<br>";
echo "Before update, myItem inventory: " . $myItem->getInventory() . "<br>";
$myItem->setPrice(20.99);
$myItem->setInventory(10);
echo "After update, myItem price USD: " . $myItem->getPrice() . "<br>";
echo "After update, myItem inventory: " . $myItem->getInventory() . "<br>";
```

- Currently, **\$itemsArr** is a multi-dimensional array that contains **3 items**. It is an **associative array** of associative arrays.
- Create a **new indexed array of Item objects** such that it contains the same information. For example, one **Item Object** will look like this:

<u>myItem:Item</u>	
id	A001
desc	Supreme Shampoo
price	5.75
inventory	12
+ info()	

- Name this **new indexed array** **\$itemObjectsArr**
- It must contain **3 Item objects**

- Update **display.php** such that it pulls **item data** from **\$itemObjectsArr** that you just created.
- **display.php** must function the same way – though the way **items** are represented changed in the background (we now use **Classes/Objects** instead of **multi-dimensional array**)



Constants

- Defined using keyword `const`

- **SalesItem.php** -

```
<?php
    class SalesItem {
        private $amount;
        const GST_RATE = 0.07;
        ...
    }
?>
```

- How to use a constant?
 - Inside the Class
`self::GST_RATE`
 - From outside the Class
`SalesItem::GST_RATE`

Exercises

- Try **Weeks 9-10 Exercises:**

- <https://smu.sg/2021-spring-is113-wk9-doc> (Google Doc)

- Complete

- **Question 1** (Car)
 - **Question 2** (Drinks)
 - **Question 3** (Dating)
 - **Question 4** (Quotes)
 - **Question 5** (Basket)
 - **Question 6** (Quotes & Likes)