# Lab: Objects and Classes

Problems for lab for the ["PHP Fundamentals" course @ SoftUni](https://softuni.bg/trainings/2344/php-fundamentals-may-2019).

You can check your solutions in [Judge](https://judge.softuni.bg/Contests/1221/).

## Day of Week

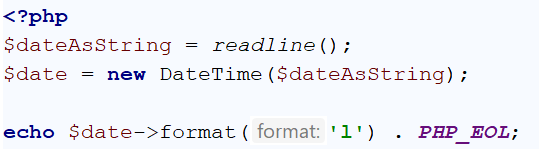
You are given a **date** in format "{day}-{month}-{year}". Calculate and print the **day of week** in **English**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 18-04-2016 | Monday |
| 27-11-1996 | Wednesday |

### Hints

* **Read the date as string** from the Console.
* Use DateTime() to convert the input string to object of typeDateTime
* Format the received date with **'l'** - this will return a full textual representation of the day of the week  
  You can see [here](http://php.net/manual/en/datetime.createfromformat.php) for more details



## Person Info

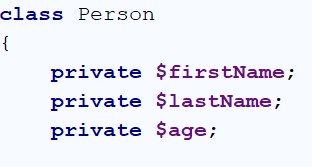
Create a person class that receives **first name**, **last name** and **age**. Print the entries of a given object.

### Examples

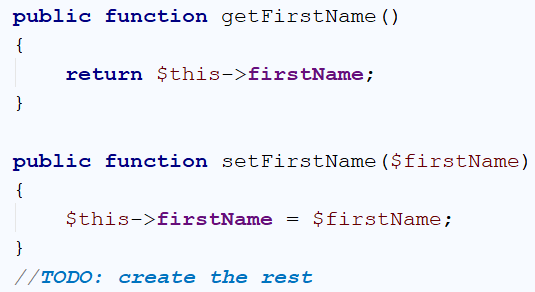
|  |  |
| --- | --- |
| **Input** | **Output** |
| Peter  Pan  20 | firstName: Peter  lastName: Pan  age: 20 |

### Hints

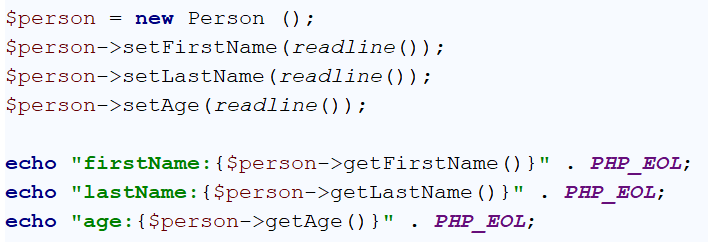
* Create the Person class and create fields



* Create getters and setters for firstName, lastName and age



* Create new Person Object and print the result



## Songs

Define a class Song, which holds the following information about songs: **Type List**, **Name** and **Time**.

On the first line you will receive the **number of songs** **-** **N**.

On the **next N-lines** you will be receiving data in the following format: "{typeList}\_{name}\_{time}"**.**

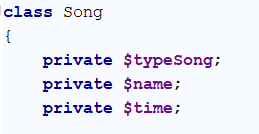
On the last line you will receive **"{Type List}"** or **"all"**. Print only the **Names of the songs** which are from that  
**"{Type List}"** or **"all"**.

### Examples

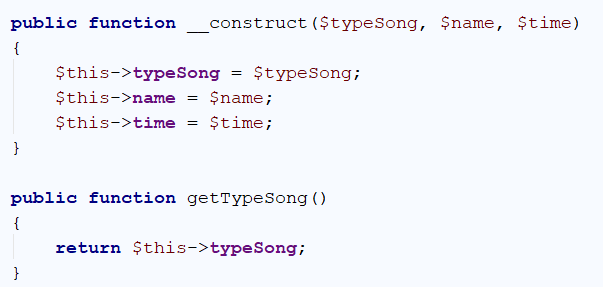
|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  favourite\_DownTown\_3:14  favourite\_Kiss\_4:16  favourite\_Smooth Criminal\_4:01  favourite | DownTown  Kiss  Smooth Criminal |
| 4  favourite\_DownTown\_3:14  listenLater\_Andalouse\_3:24  favourite\_In To The Night\_3:58  favourite\_Live It Up\_3:48  listenLater | Andalouse |
| 2  like\_Replay\_3:15  ban\_Photoshop\_3:48  all | Replay  Photoshop |

### Solution

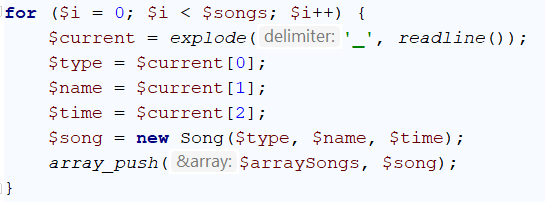
Define class Song with properties: **Type List**, **Name** and **Time**.



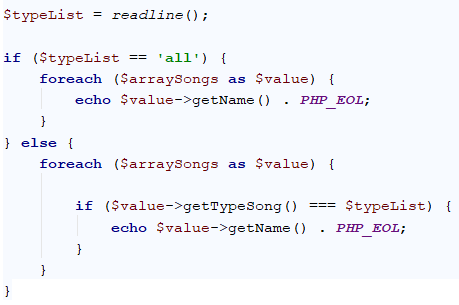
Create constructor and getters:



Read the input lines, make collection and store the data.



Finally read your last line – **Type List** and **print** the result.



## Students

Define a class Student, which holds the following information about students: **first name**, **last name**, **age** and **hometown**.

Read list of students until you receive "**end**" command. After that, you will receive a **city name**. Print only students which are from the given city, in the following format: **"{**firstName} {lastName**} is {age} years old."**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| John Smith 15 Sofia  Peter Ivanov 14 Plovdiv  Linda Bridge 16 Sofia  Simon Stone 12 Varna  end  Sofia | John Smith is 15 years old.  Linda Bridge is 16 years old. |
| Anthony Taylor 15 Chicago  David Anderson 16 Washington  Jack Lewis 14 Chicago  David Lee 14 Chicago  end  Chicago | Anthony Taylor is 15 years old.  Jack Lewis is 14 years old.  David Lee is 14 years old. |

### Hints

* Define a class Student with the following fields: firstName, lastName, age and city.
* Generate constructor in class Student.
* Read a list of students.
* Read a city name and print only students which are from the given city.

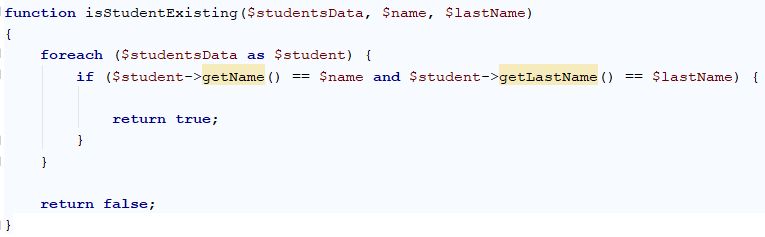
## Students 2.0

Use the class from the previous problem. If you receive a student which already exists (**first name** and **last name** should be **unique**) overwrite the information.

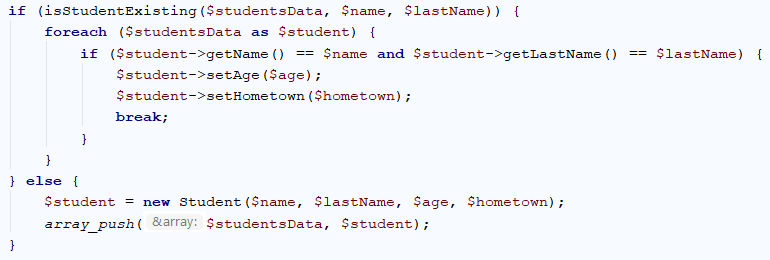
|  |  |
| --- | --- |
| **Input** | **Output** |
| John Smith 15 Sofia  Peter Ivanov 14 Plovdiv  Peter Ivanov 25 Plovdiv  Linda Bridge 16 Sofia  Linda Bridge 27 Sofia  Simon Stone 12 Varna  end  Sofia | John Smith is 15 years old.  Linda Bridge is 27 years old. |
| Anthony Taylor 15 Chicago  David Anderson 16 Washington  Jack Lewis 14 Chicago  David Lee 14 Chicago  Jack Lewis 26 Chicago  David Lee 18 Chicago  end  Chicago | Anthony Taylor is 15 years old.  Jack Lewis is 26 years old.  David Lee is 18 years old. |

### Hints

Check if the given student already exists.



Overwrite the student information if the student exists.



## Store Boxes

Define a class **Item** which contains these properties: **Name and Price.**

Define a class **Box** which contains these properties: **Serial Number, Item, Item Quantity and Price for a Box.**

Until you receive **"end"** you will be receiving data in the following format:  
"{Serial Number} {Item Name} {Item Quantity} {itemPrice}"

The **Price of one box** has to be calculated: itemQuantity \* itemPrice**.**

Print all the boxes, ordered descending by price for a box, in the following format:

{boxSerialNumber}

-- {boxItemName} – ${boxItemPrice}: {boxItemQuantity}

-- ${boxPrice}

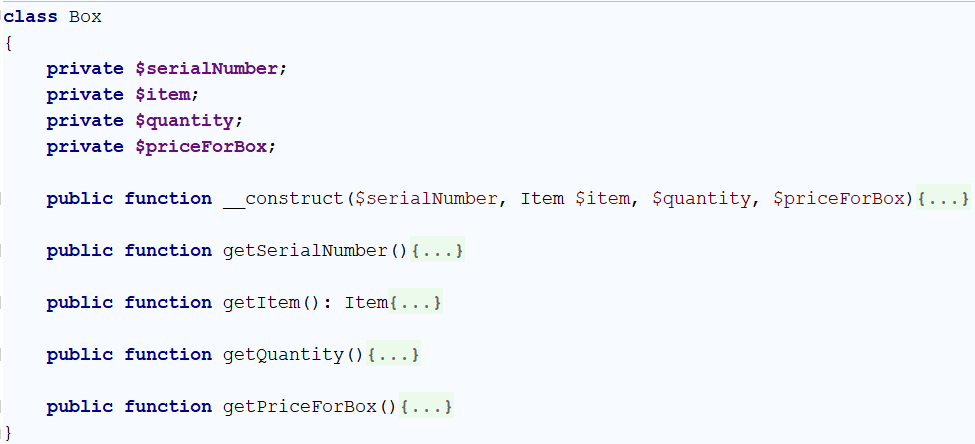
Price should be **formatted to the 2nd character after the decimal point**.

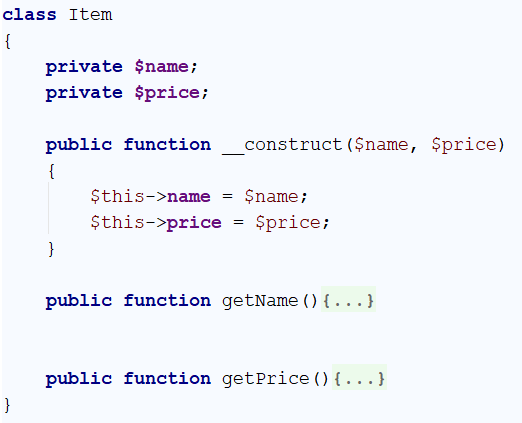
### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 19861519 Dove 15 2.50  86757035 Butter 7 3.20  39393891 Orbit 16 1.60  37741865 Samsung 10 1000  end | 37741865  -- Samsung - $1000.00: 10  -- $10000.00  19861519  -- Dove - $2.50: 15  -- $37.50  39393891  -- Orbit - $1.60: 16  -- $25.60  86757035  -- Butter - $3.20: 7  -- $22.40 |
| 48760766 Alcatel 8 100  97617240 Intel 2 500  83840873 Milka 20 2.75  35056501 SneakersXL 15 1.50  end | 97617240  -- Intel - $500.00: 2  -- $1000.00  48760766  -- Alcatel - $100.00: 8  -- $800.00  83840873  -- Milka - $2.75: 20  -- $55.00  35056501  -- SneakersXL - $1.50: 15  -- $22.50 |

### Hints

This is how your class Box and class Items should look like.





## Vehicle Catalogue

Your task is to **create Vehicle catalogue** which contains only **Trucks** **and** **Cars**.

Define class **Truck** with these fields: **Brand, Model and Weight**.

Define class **Car** with these fields: **Brand, Model and Horse Power**.

Define class **Catalog** with these fields: **Collections of** **Trucks and Cars**.

You have to read your input until you receive the "**end**" command.

The input will be in following format: "{type}/{brand}/{model}/{horse power / weight}"

In the end you have **to print all vehicles ordered alphabetical by brand,** in the following format:

Cars:

{Brand}: {Model} - {Horse Power}hp

Trucks:

{Brand}: {Model} - {Weight}kg

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Car/Audi/A3/110  Car/Maserati/Levante/350  Truck/Mercedes/Actros/9019  Car/Porsche/Panamera/375  end | Cars:  Audi: A3 - 110hp  Maserati: Levante - 350hp  Porsche: Panamera - 375hp  Trucks:  Mercedes: Actros - 9019kg |
| Car/Subaru/Impreza/152  Car/Peugeot/307/109  end | Cars:  Peugeot: 307 - 109hp  Subaru: Impreza - 152hp |

### Hints

This is how your class **Catalog** should look like.

