**PHP- Advance PHP**

MODULE -4

(PHP Assignment)

Advance PHP

**Submitted To:-**

**Rajesh Nagar Sir**

**Submitted By:-**

**Ankita Sahu**

**MODULE – 4(Advance PHP)**

OOPS:

**1. What Is Object Oriented Programming?**

**Ans:- Object-oriented programming** **(OOPs) has a sweeping impact because it appeals at multiple levels and promises faster and cheaper development and maintenance. It follows a bottom-up approach to develop applications.** **The word object-oriented is the combination of two words i.e. object and oriented. The dictionary meaning of the object is an article or entity that exists in the real world. The meaning of oriented is interested in a particular kind of thing or entity. In layman's terms, it is a programming pattern that rounds around an object or entity are called object-oriented programming.**

**2. What Are Properties Of Object Oriented Systems?**

**Ans:- There are some basic Properties Of Object Oriented Systems (OOPs)**

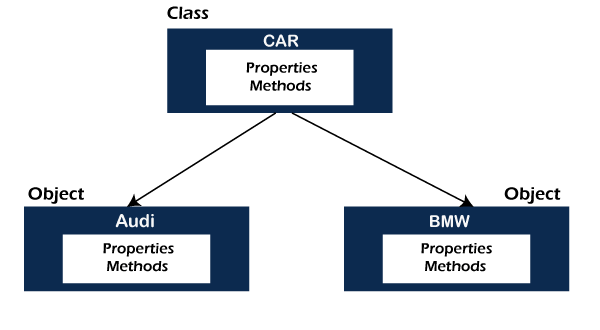
* **Objects**
* **Class**
* **Encapsulation**
* **Abstraction**
* **Polymorphism**
* **Inheritance**

**Object:**

**An**[**object**](https://www.javatpoint.com/object-and-class-in-java#object)**is a real-world entity that has attributes, behavior, and properties. It is referred to as an instance of the class. It contains member functions, variables that we have defined in the class. It occupies space in the memory. Different objects have different states or attributes, and behaviors.**

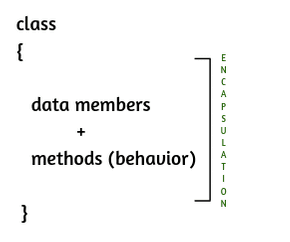
### Class:

**A**[**class**](https://www.javatpoint.com/object-and-class-in-java#class)**is a blueprint or template of an object. It is a user-defined data type. Inside a class, we define variables, constants, member functions, and other functionality. it binds data and functions together in a single unit. It does not consume memory at run time. Note that classes are not considered as a data structure. It is a logical entity. It is the best example of data binding. Note that a class can exist without an object but vice-versa is not possible.**

****

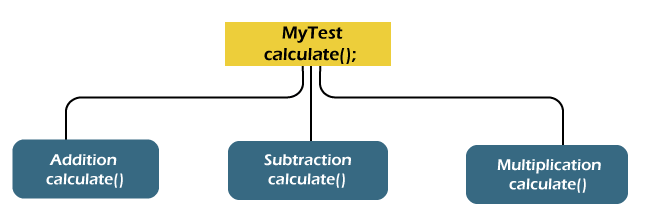
### Encapsulation:

[**Encapsulation**](https://www.javatpoint.com/encapsulation)**is a mechanism that allows us to bind data and functions of a class into an entity. It protects data and functions from outside interference and misuse. Therefore, it also provides security. A class is the best example of encapsulation.**

****

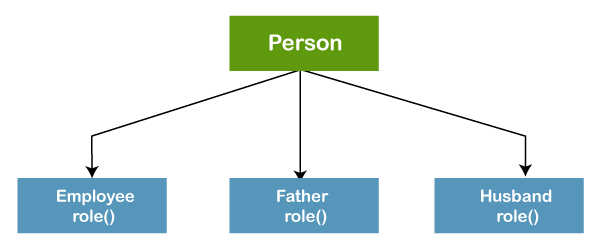
### Abstraction:

**The concept allows us to hide the implementation from the user but shows only essential information to the user. Using the concept developer can easily make changes and added over time.**

****

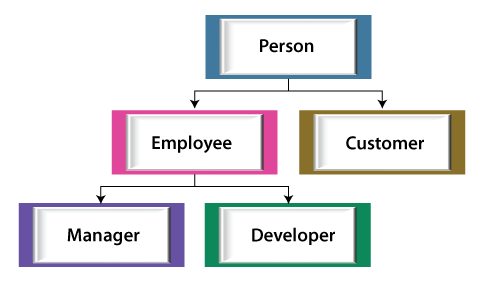
### Polymorphism:

**The word**[**polymorphism**](https://www.javatpoint.com/runtime-polymorphism-in-java)**is derived from the two words i.e. ploy and morphs. Poly means many and morphs means forms. It allows us to create methods with the same name but different method signatures. It allows the developer to create clean, sensible, readable, and resilient code.**

****

### Inheritance:

**The concept allows us to inherit or acquire the properties of an existing class (parent class) into a newly created class (child class). It is known as**[**inheritance**](https://www.javatpoint.com/inheritance-in-java)**. It provides code reusability.**

****

**3. What Is Difference Between Class And Interface?**

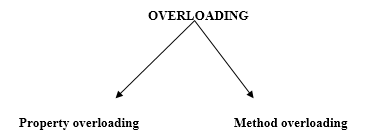
**Ans:-**

1. **Definition:**
   * **Class:** A class in PHP is a blueprint for creating objects. It defines the properties (variables) and methods (functions) that objects of the class will have.
   * **Interface:** An interface defines a contract that classes can adhere to. It specifies a set of methods that implementing classes must define. An interface itself cannot contain any implementation details, only method signatures.
2. **Inheritance:**
   * **Class:** Classes can use inheritance to inherit properties and methods from a parent class. A class can extend only one parent class in PHP due to its single inheritance nature.
   * **Interface:** Interfaces do not support inheritance in the same way as classes. A class can implement multiple interfaces, allowing it to provide implementations for all the methods specified in those interfaces.
3. **Method Implementation:**
   * **Class:** Classes can provide full method implementations for the methods they define.
   * **Interface:** Interfaces only declare method signatures without any implementation. Any class implementing an interface must provide implementations for all the methods declared in the interface.
4. **Instantiation:**
   * **Class:** Objects are instantiated from classes using the **new** keyword. The class can have a constructor method to initialize object properties.
   * **Interface:** Interfaces cannot be instantiated directly. They are used to define a contract that classes can adhere to.
5. **Properties:**
   * **Class:** Classes can have properties (variables) that store data related to the object.
   * **Interface:** Interfaces cannot have properties; they only declare method signatures.
6. **Usage:**
   * **Class:** Classes are used to create objects and define their behavior. They can have methods with logic and properties to store data.
   * **Interface:** Interfaces are used to define a common set of methods that multiple classes should implement. They allow for polymorphism and a clear separation of concerns.
7. **Abstract Classes:**
   * **Class:** Abstract classes are classes that cannot be instantiated on their own. They can contain both abstract methods (methods without implementation) and concrete methods (methods with implementation). Abstract classes can also provide a partial implementation of a class, allowing subclasses to complete the implementation.
   * **Interface:** Interfaces are purely a contract for methods; they cannot contain any implementation. However, a class can implement multiple interfaces and extend one abstract class simultaneously.

**4. What Is Overloading?**

**Ans:-**

* Overloading in **PHP** provides means to dynamically **create properties and methods**.
* **These dynamic entities** are processed via magic methods, one can establish in a class for various action types.
* **All** overloading methods must be defined as **Public**.
* After creating object for a class, we can access set of entities that are properties or methods not defined within the scope of the class.
* Such entities are said to be **overloaded properties or methods**, and the process is called as overloading.
* For working with these overloaded properties or functions, **PHP magic methods are used**.
* Most of the magic methods will be triggered in object context except \_\_callStatic() method which is used in static context.



## Property overloading:

* PHP property overloading allows us to create dynamic properties in object context.
* For creating those properties no separate line of code is needed.
* A property which is associated with class instance, and not declared within the scope of the class, is considered as overloaded property.

**Some of the magic methods which is useful for property overloading:-**

* **\_\_set():** It is triggered while initializing overloaded properties.
* **\_\_get():** It is utilized for reading data from inaccessible Properties.
* **\_\_isset():** This magic method is invoked when we check overloaded properties with isset() function.
* **\_\_unset():** This function will be invoked on using PHP unset() for overloaded properties.

## Method overloading:

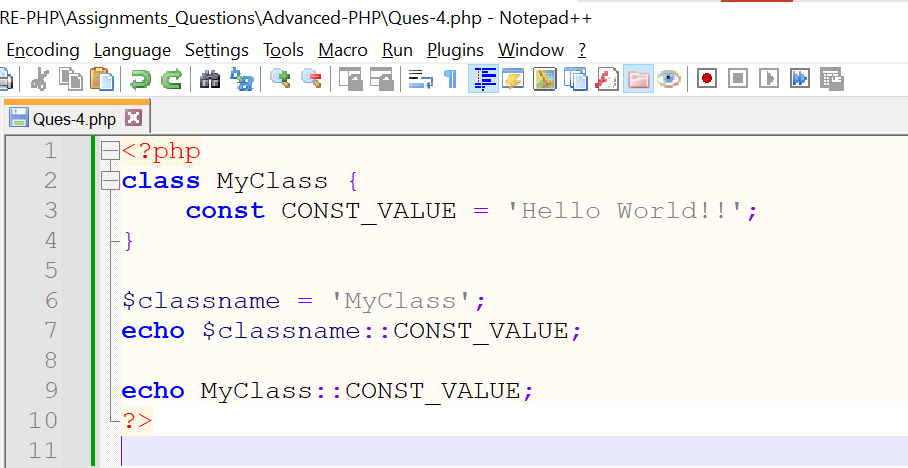
Method Overloading is a concept of Object Oriented Programming which helps in building the composite application in an easy way. Function overloading or method overloading is a feature that permits making creating several methods with a similar name that works differently from one another in the type of the input parameters it accepts as arguments.

**5. What Is T\_PAAMAYIM\_NEKUDOTAYIM (Scope Resolution Operator (::) with Example?**

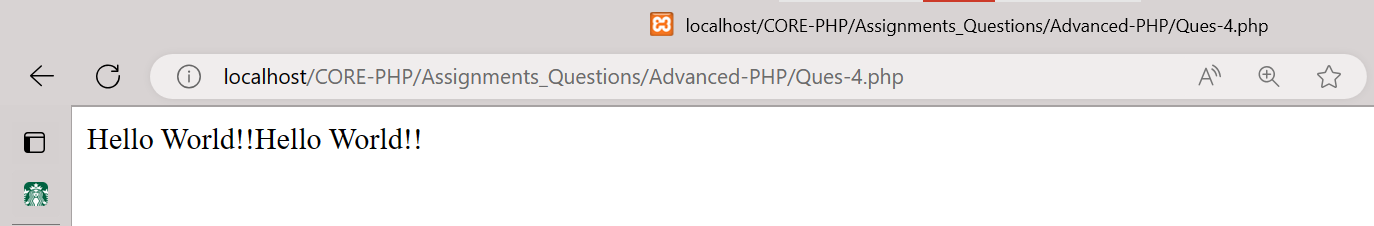
**Ans:-**

In PHP, the double colon**::**is defined as **Scope Resolution Operator**. It used when when we want to access constants, properties and methods defined at class level. When referring to these items outside class definition, name of class is used along with scope resolution operator. This operator is also called **Paamayim Nekudotayim**, which in Hebrew means double colon.

**Code:**

****

**Output:**

****

**6. What are the differences between abstract classes and interfaces?**

**Ans:-**

|  |  |
| --- | --- |
| **Abstract class** | **Interface** |
| 1) Abstract class can **have abstract and non-abstract** methods. | Interface can have **only abstract** methods. Since Java 8, it can have **default and static methods** also. |
| 2) Abstract class **doesn't support multiple inheritance**. | Interface **supports multiple inheritance**. |
| 3) Abstract class **can have final, non-final, static and non-static variables**. | Interface has **only static and final variables**. |
| 4) Abstract class **can provide the implementation of interface**. | Interface **can't provide the implementation of abstract class**. |
| 5) The **abstract keyword** is used to declare abstract class. | The **interface keyword** is used to declare interface. |
| 6) An **abstract class** can extend another Java class and implement multiple Java interfaces. | An **interface** can extend another Java interface only. |
| 7) An **abstract class** can be extended using keyword "extends". | An **interface** can be implemented using keyword "implements". |
| 8) A Java **abstract class** can have class members like private, protected, etc. | Members of a Java interface are public by default. |
| 9)**Example:** public abstract class Shape{ public abstract void draw(); } | **Example:** public interface Drawable{ void draw(); } |

**7. Define Constructor and Destructor?**

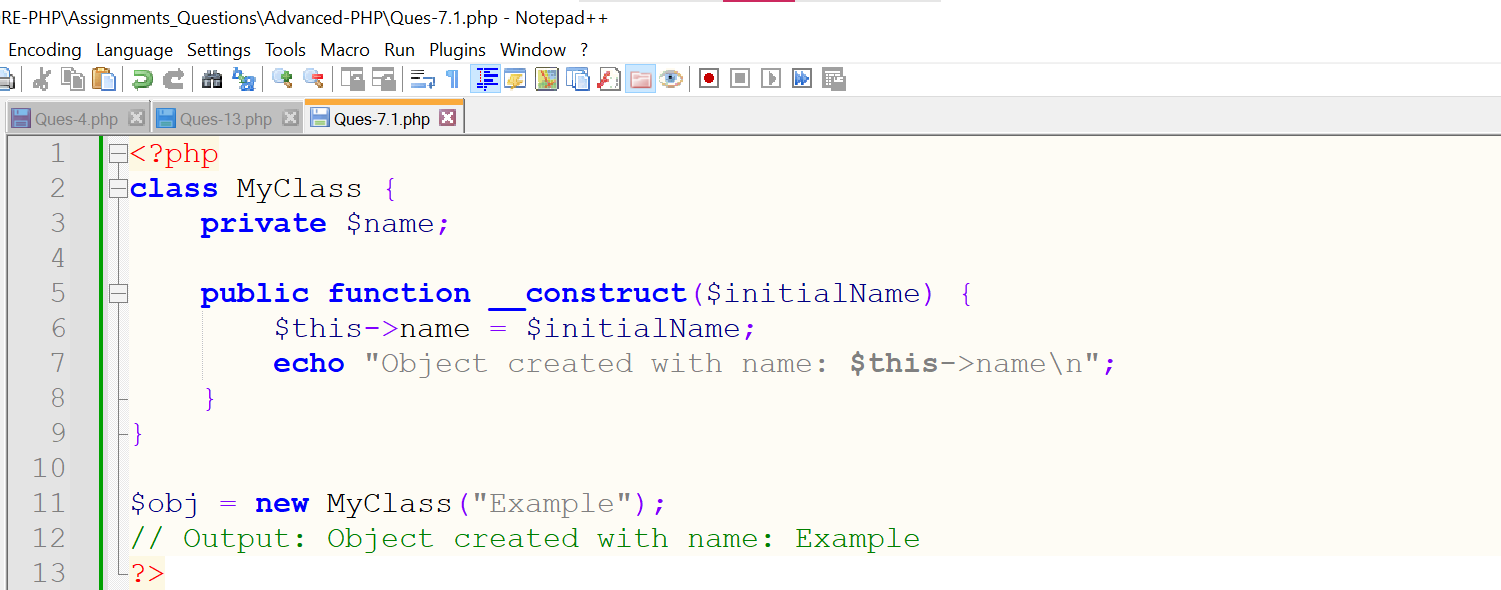
**Ans:-**

**Constructor:-**

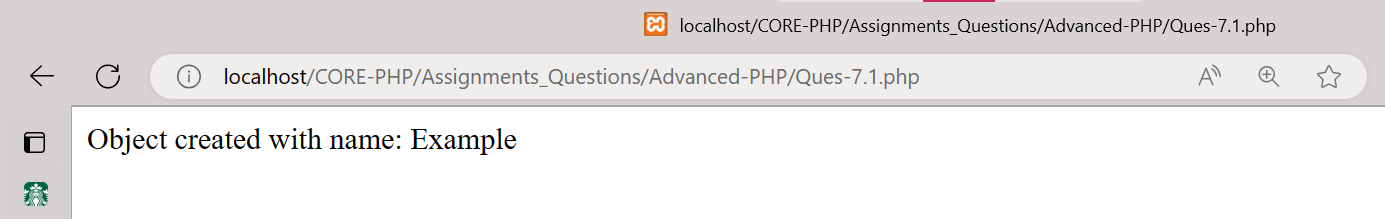
A constructor is a special method within a class that is automatically called when an object of that class is created. It is used to initialize the object's properties or perform any necessary setup tasks. In PHP, the constructor method is named **\_\_construct().**

**An example of a constructor in PHP:**

**Code:**

****

**Output:**

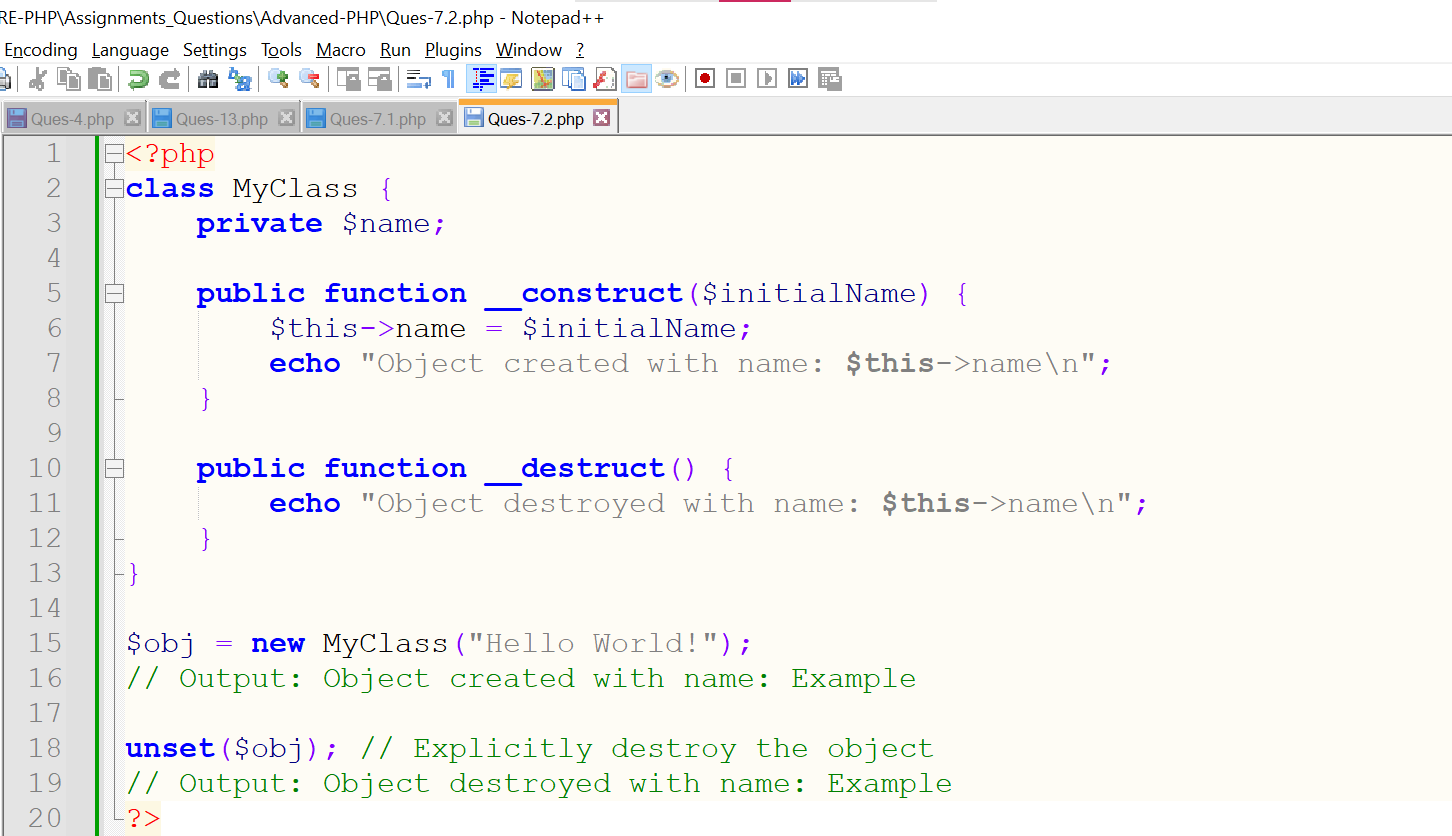
****

**Destructor:-**

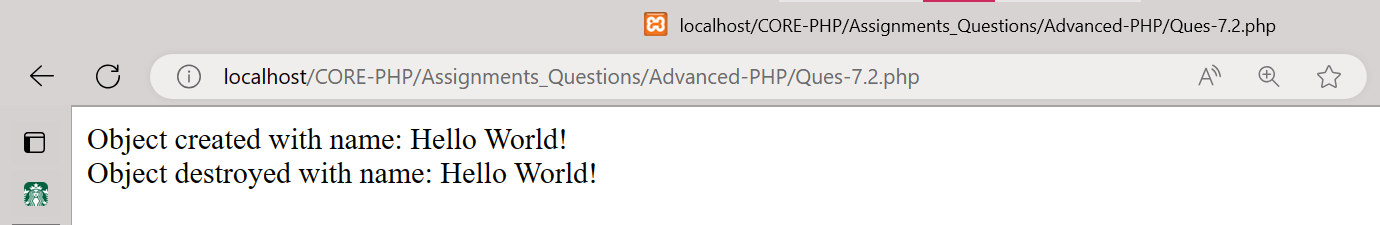
A destructor is a special method within a class that is automatically called when an object of that class is no longer referenced or explicitly destroyed. It is used to perform cleanup tasks before the object is removed from memory. In PHP, the destructor method is named **\_\_destruct().**

An example of a destructor in PHP:

**Code:**

****

**Output:**

****

**8. How to Load Classes in PHP?**

**Ans:-** PHP load classes are used for declaring its object etc. in object oriented applications. PHP parser loads it automatically, if it is registered with **spl\_autoload\_register()**function. PHP parser gets the least chance to load class/interface before emitting an error.

**Syntax:**

spl\_autoload\_register(function ($class\_name)

{

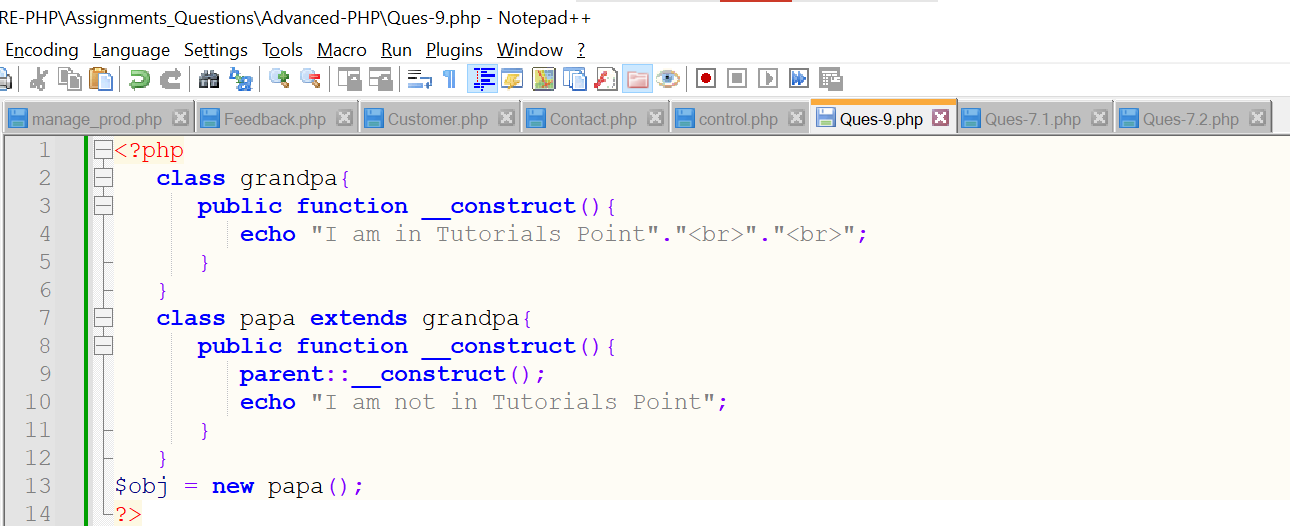
include $class\_name . '.php';

});

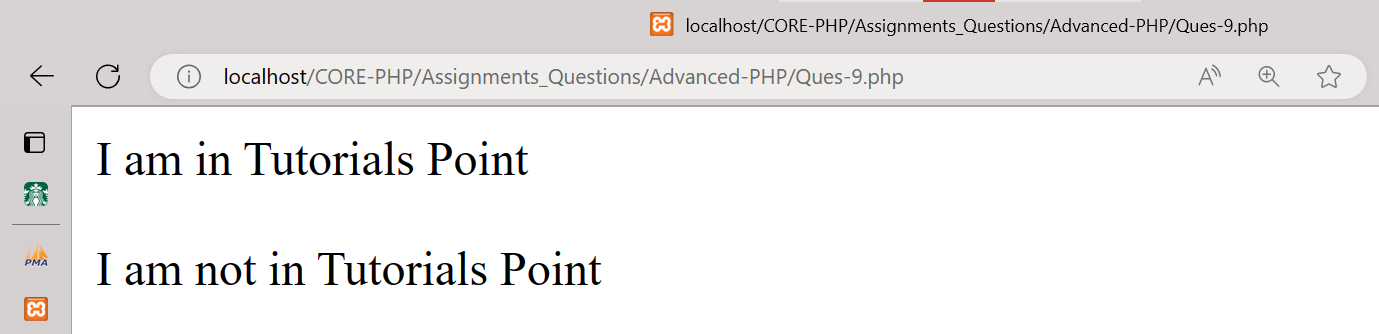
**9. How to Call Parent Constructor?**

**Ans:-** We can't run directly the parent class constructor in child class if the child class defines a constructor. In order to run a parent constructor, a call to parent::\_\_construct() within the child constructor is required.

**Code:**



**Output:**

****

**10. Are Parent Constructor Called Implicitly When Create An Object Of Class?**

**Ans:-** Parent constructors are not called implicitly if the child class defines a constructor. In order to run a parent constructor, a call to **parent::\_\_construct().**

**11. What Happen, If Constructor Is Defined As Private Or Protected?**

**Ans:-** The constructor may be made private or protected to prevent it from being called externally. If so, only a static method will be able to instantiate the class. Because they are in the same class definition they have access to private methods, even if not of the same object instance.

**12. What are PHP Magic Methods/Functions? List them Write program for Static Keyword in PHP?**

**Ans:-** Magic methods in PHP are special methods that are aimed to perform certain tasks. These methods are named with double underscore (\_\_) as prefix. All these function names are reserved and can't be used for any purpose other than associated magical functionality. Magical method in a class must be declared public. These methods act as interceptors that are automatically called when certain conditions are met.

* Following magical methods are currently available in PHP:

## i) \_\_sleep:

public \_\_sleep ( void ) : array

serialize() method in class checks if it has a function name \_\_sleep(). If so, that function is executed prior to any serialization. It is supposed to return an array with the names of all variables of that object that should be serialized.

## ii) \_\_wakeup:

public \_\_wakeup ( void ) : void

unserialize() method checks there exists a function with the magic name \_\_wakeup(). If present, this function can reconstruct any resources that the object may have.

## iii) \_\_serialize:

public \_\_serialize ( void ) : array

serialize() method also checks if the class has \_\_serialize() method. If so, it is executed prior to any serialization. It must construct and return an associative array of key/value pairs that represent the serialized form of the object.

## v) \_\_unserialize:

public \_\_unserialize ( array $data ) : void

unserialize() also checks for if \_\_unserialize() is present, and it will be passed the restored array that was returned from \_\_serialize(). It may then restore the properties of the object from that array as appropriate.

## vi) \_\_toString:

public \_\_toString ( void ) : string

The \_\_toString() method describes string representation of object. For example, what echo $obj; will print. This method must return a string.

## vii) \_\_invoke:

\_\_invoke ([ $... ] ) : mixed

This method is called when a script tries to call an object as a function.

## viii) \_\_set\_state:

static \_\_set\_state ( array $properties ) : object

This static method is called for classes exported by var\_export(). It receices one parameter which is an array containing exported properties in the form array('property' => value, ...).

## ix) \_\_debugInfo:

\_\_debugInfo ( void ) : array

This method is automatically called when var\_dump() is executed for dumping an object to get the properties that should be shown. If it isn't defined,all public, protected and private properties will be shown.

## x) \_\_set:

public \_\_set ( string $name , mixed $value ) : void

\_\_set() is run when writing data to inaccessible (protected or private) or non-existing properties.

## xi) \_\_get:

public \_\_get ( string $name ) : mixed

\_\_get() is utilized for reading data from inaccessible (protected or private) or non-existing properties.

## xii) \_\_isset:

public \_\_isset ( string $name ) : bool

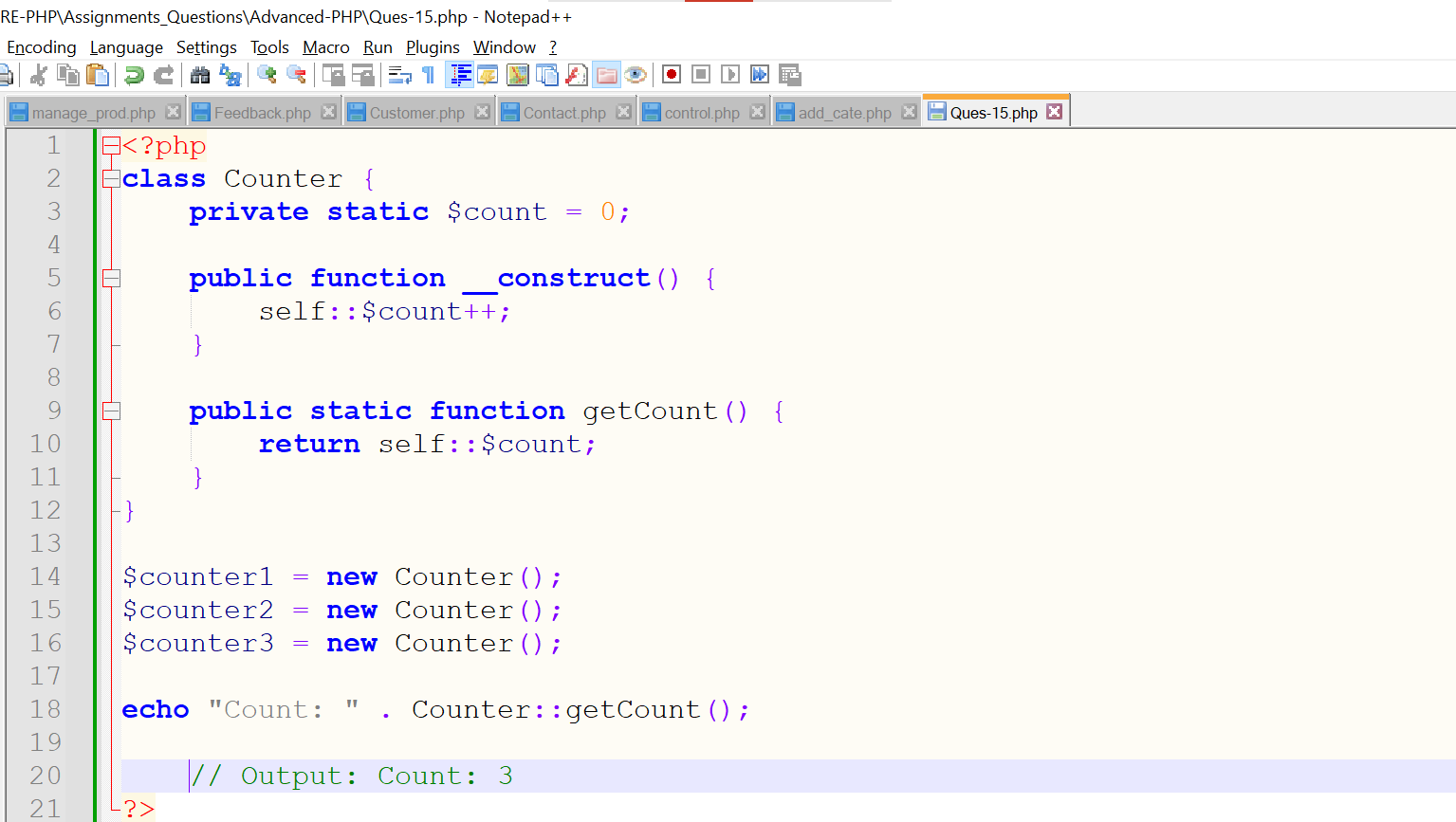
\_\_isset() is triggered by calling isset() or empty() on inaccessible (protected or private) or non-existing properties.

## xiii) \_\_unset:

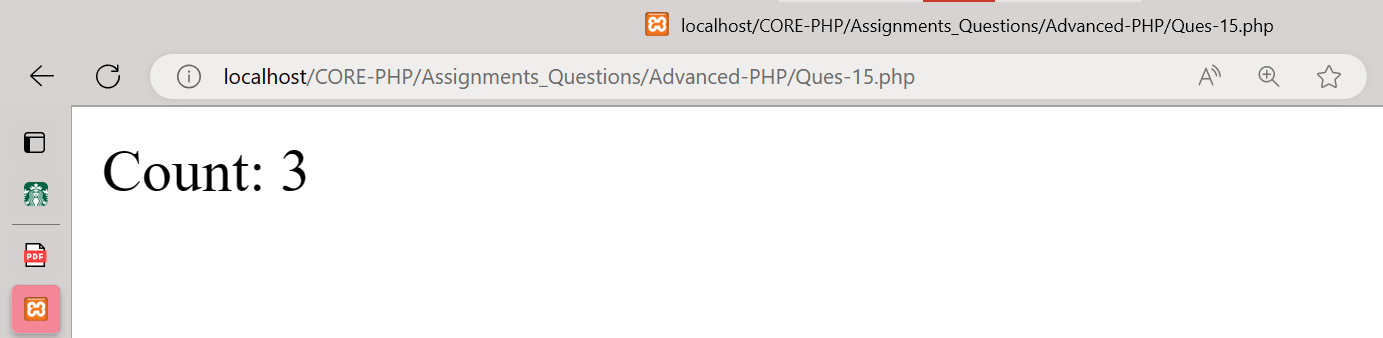
public \_\_unset ( string $name ) : void

\_\_unset() is invoked when unset() is used on inaccessible (protected or private) or non-existing properties.

**Code:**

****

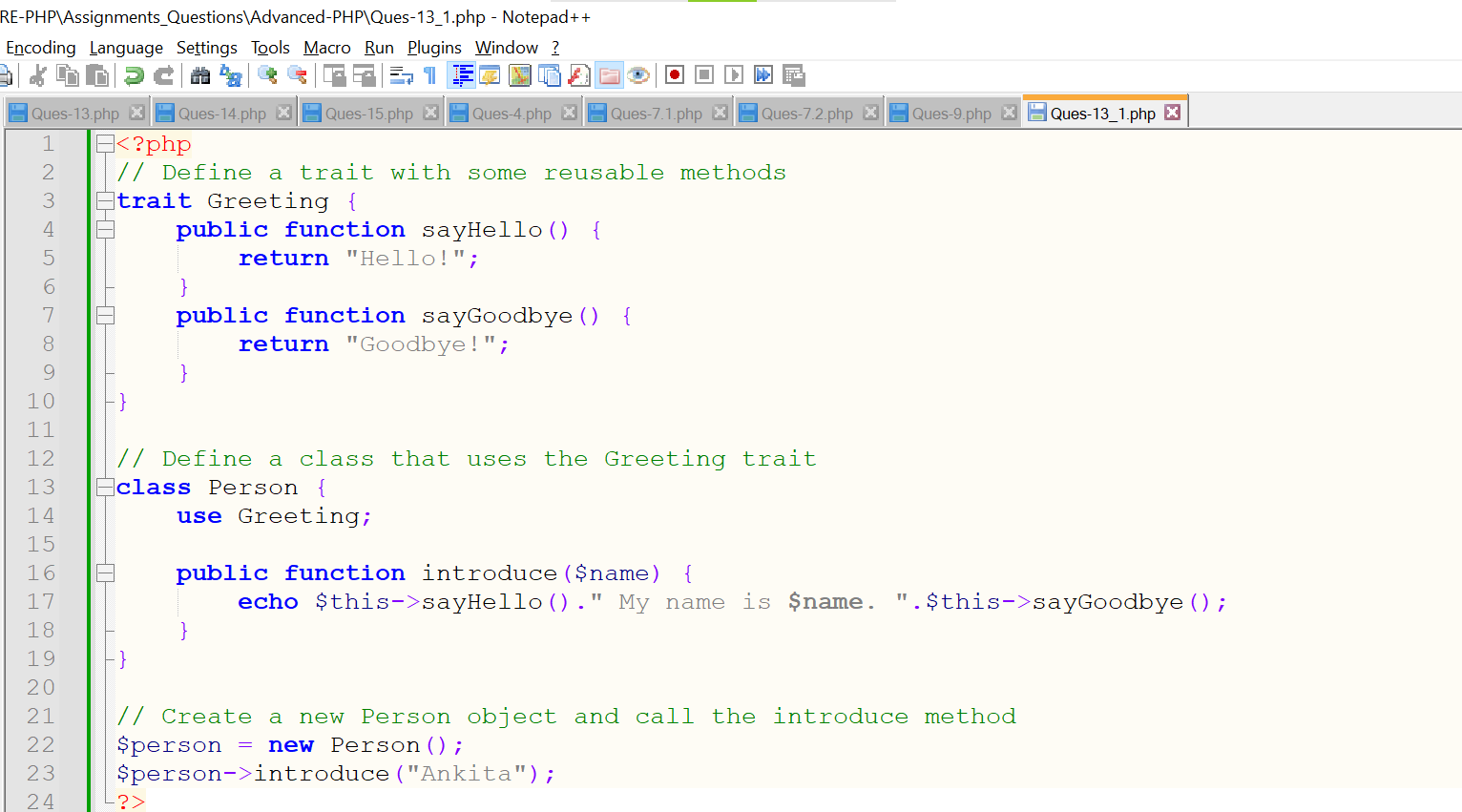
**Output:**



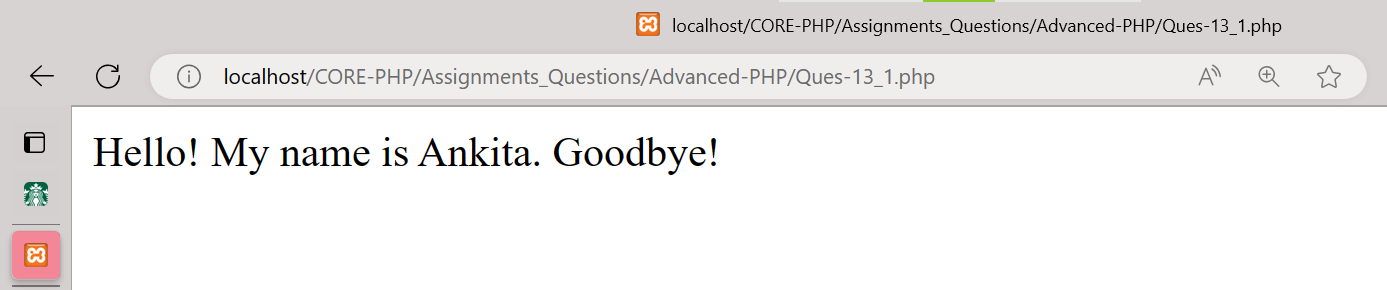
**13. Create multiple Traits and use it in to a single class?**

**Ans:-** Traits are used to declare methods that can be used in multiple classes. Traits can have methods and abstract methods that can be used in multiple classes, and the methods can have any access modifier (public, private, or protected).

**Code:**

****

**Output:**

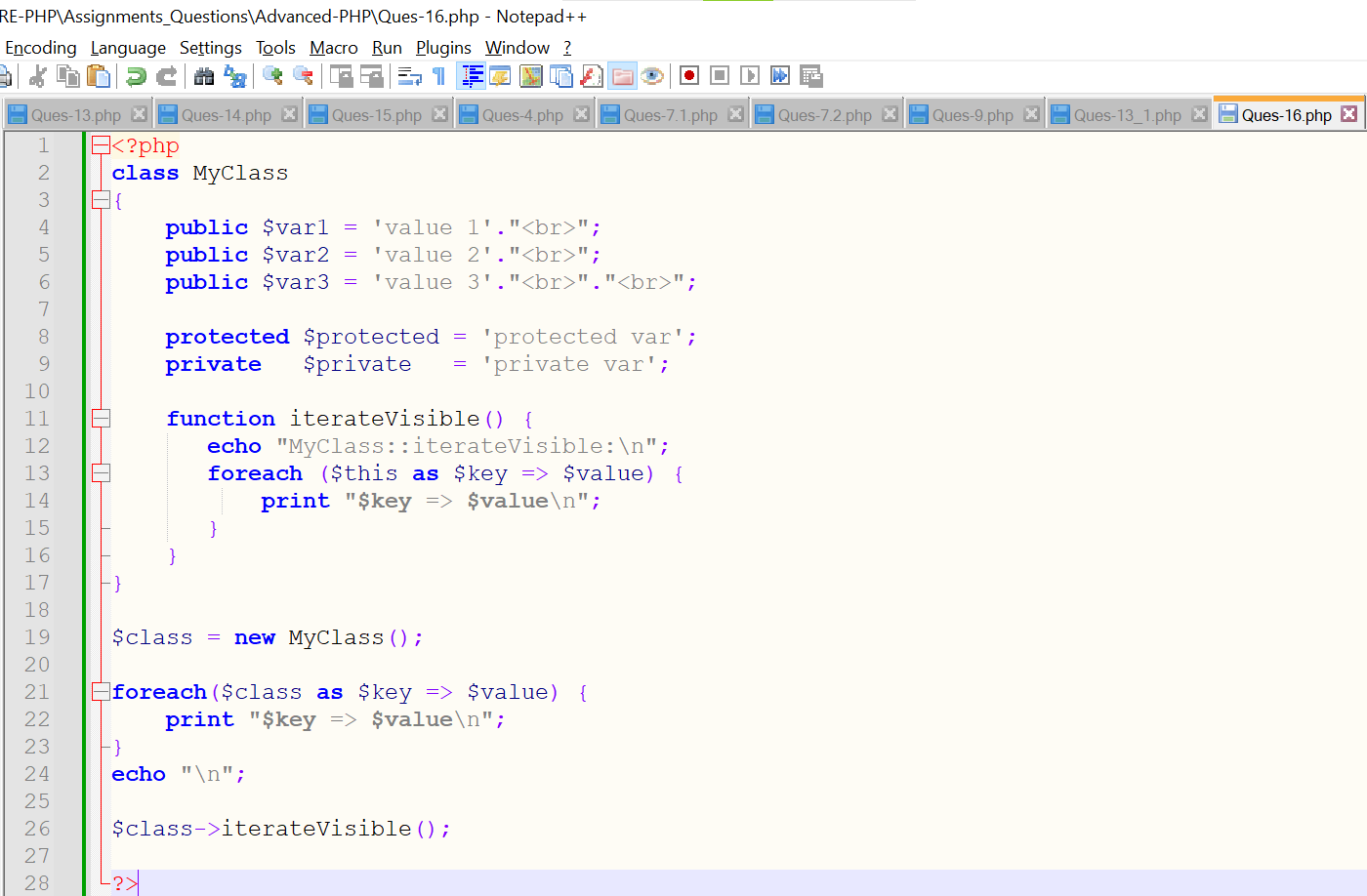


**14. Write PHP Script of Object Iteration?**

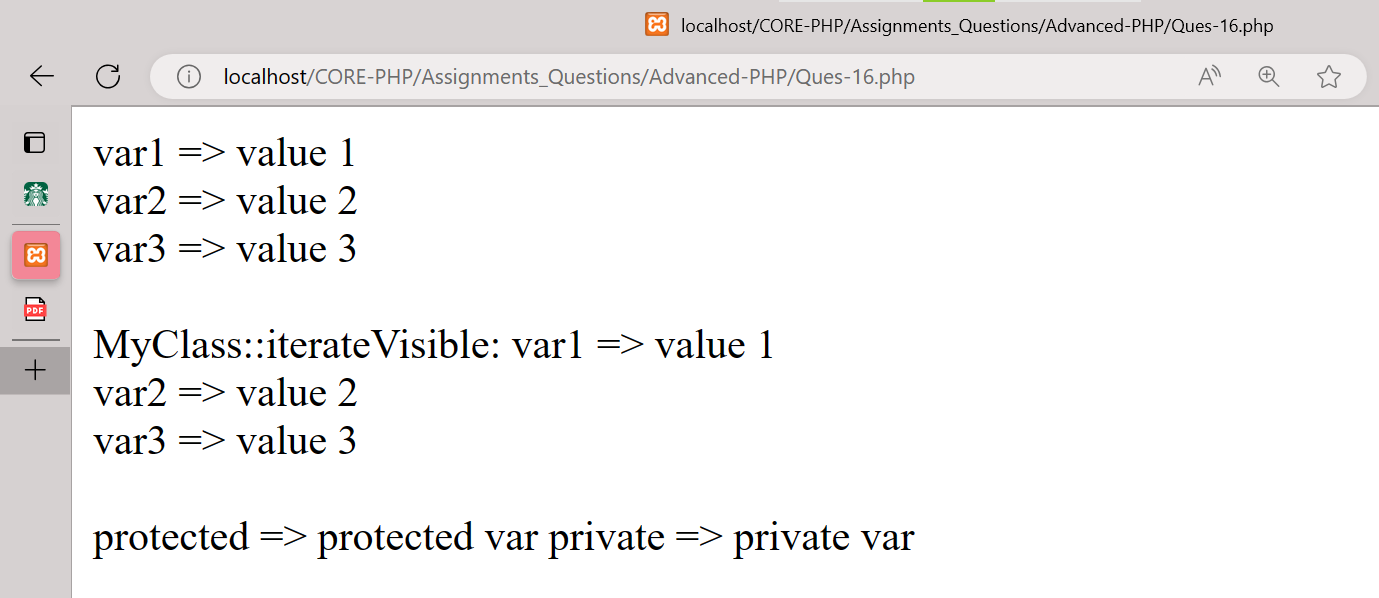
**Ans:-** When you are iterating obejcts from a list through a loop functions I.e foreach, then this process is called object iteration**.**

This concept iterates only all visible properties & methods**.**

**Code:**

****

**Output:**

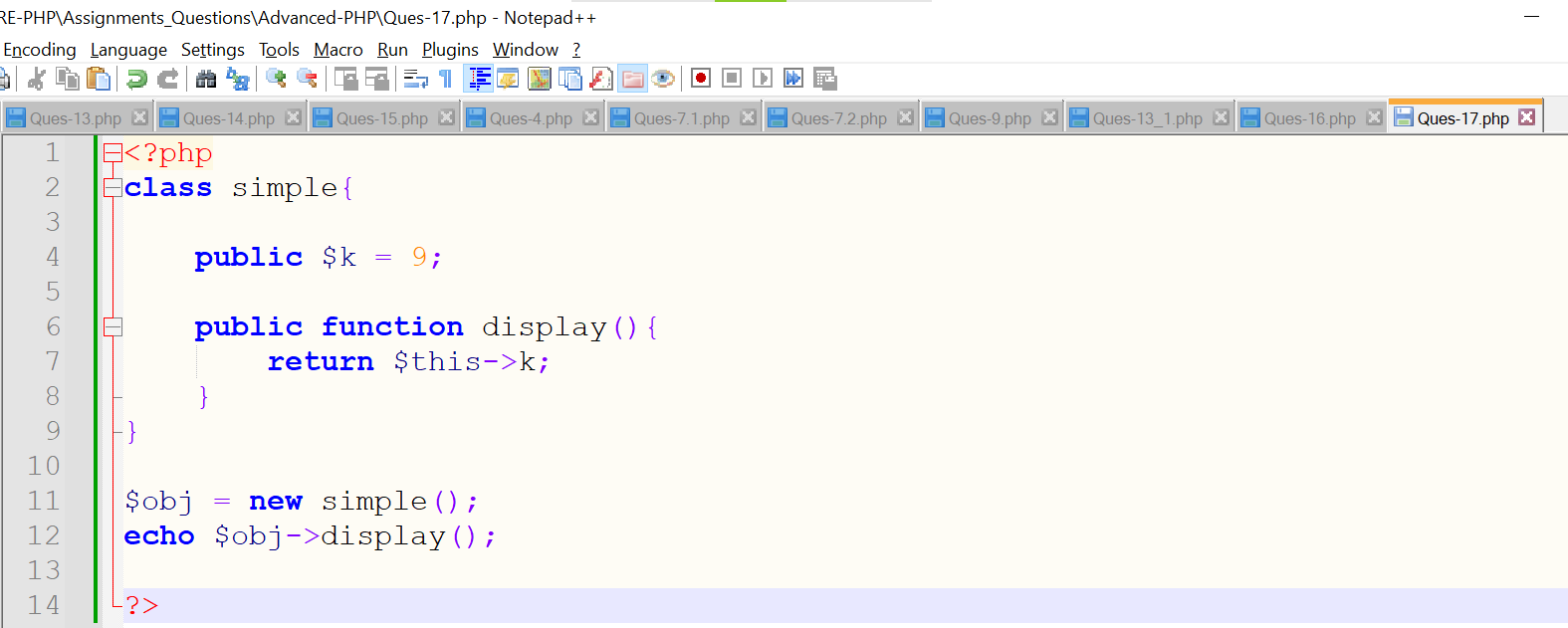
****

**15. Use of The $this keyword?**

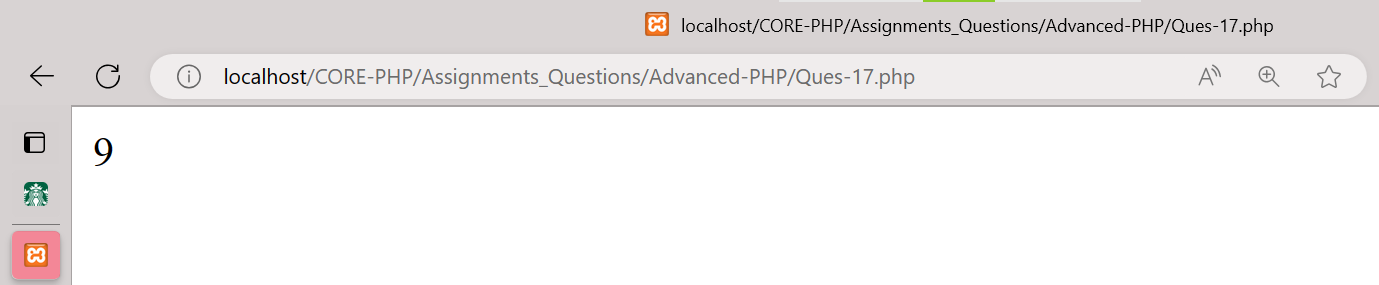
**Ans:-**

**$this** is a reserved keyword in PHP that refers to the calling object. It is usually the object to which the method belongs, but possibly another object if the method is called statically from the context of a secondary object. This keyword is only applicable to internal methods.

**Code:**

****

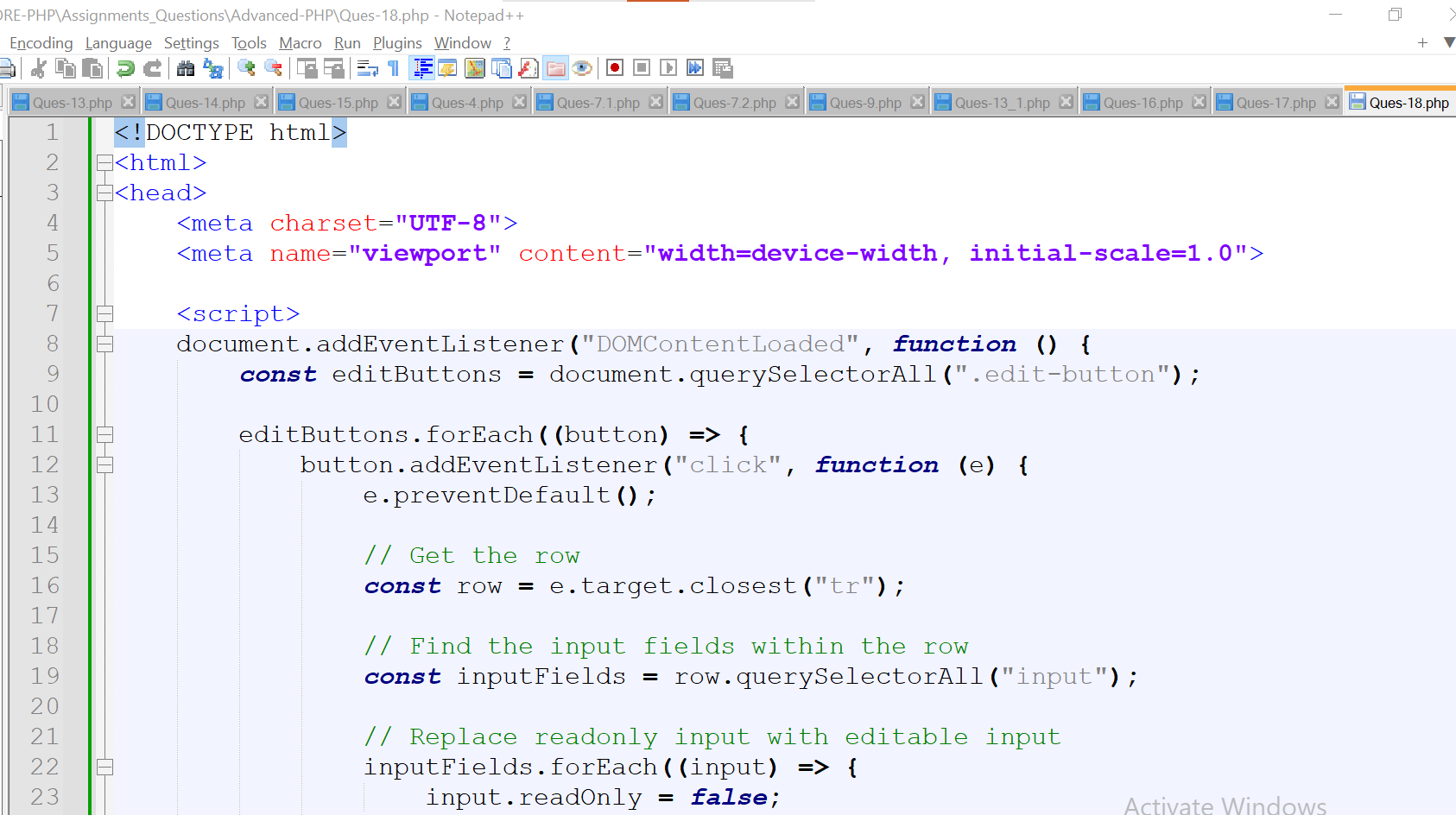
**Output:**

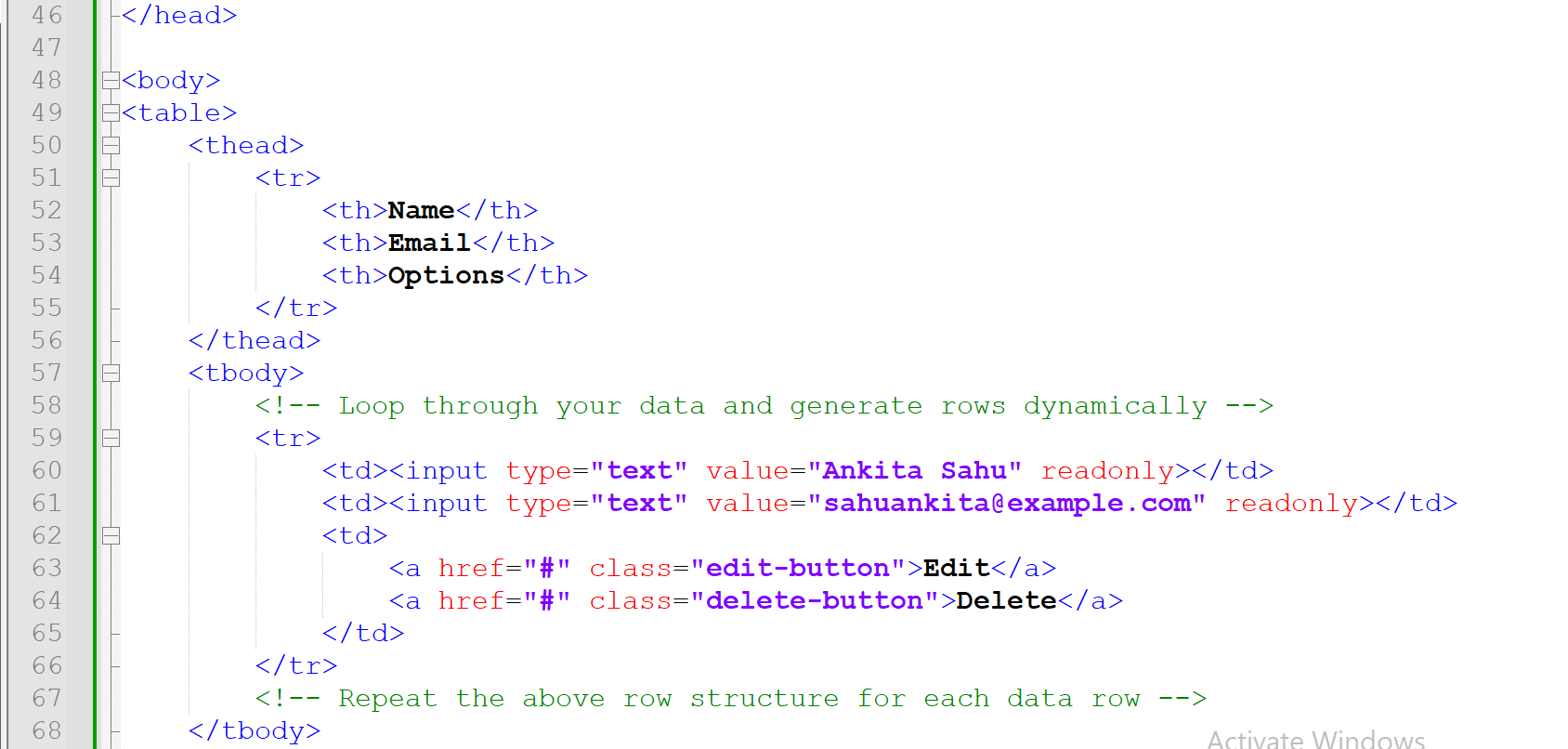
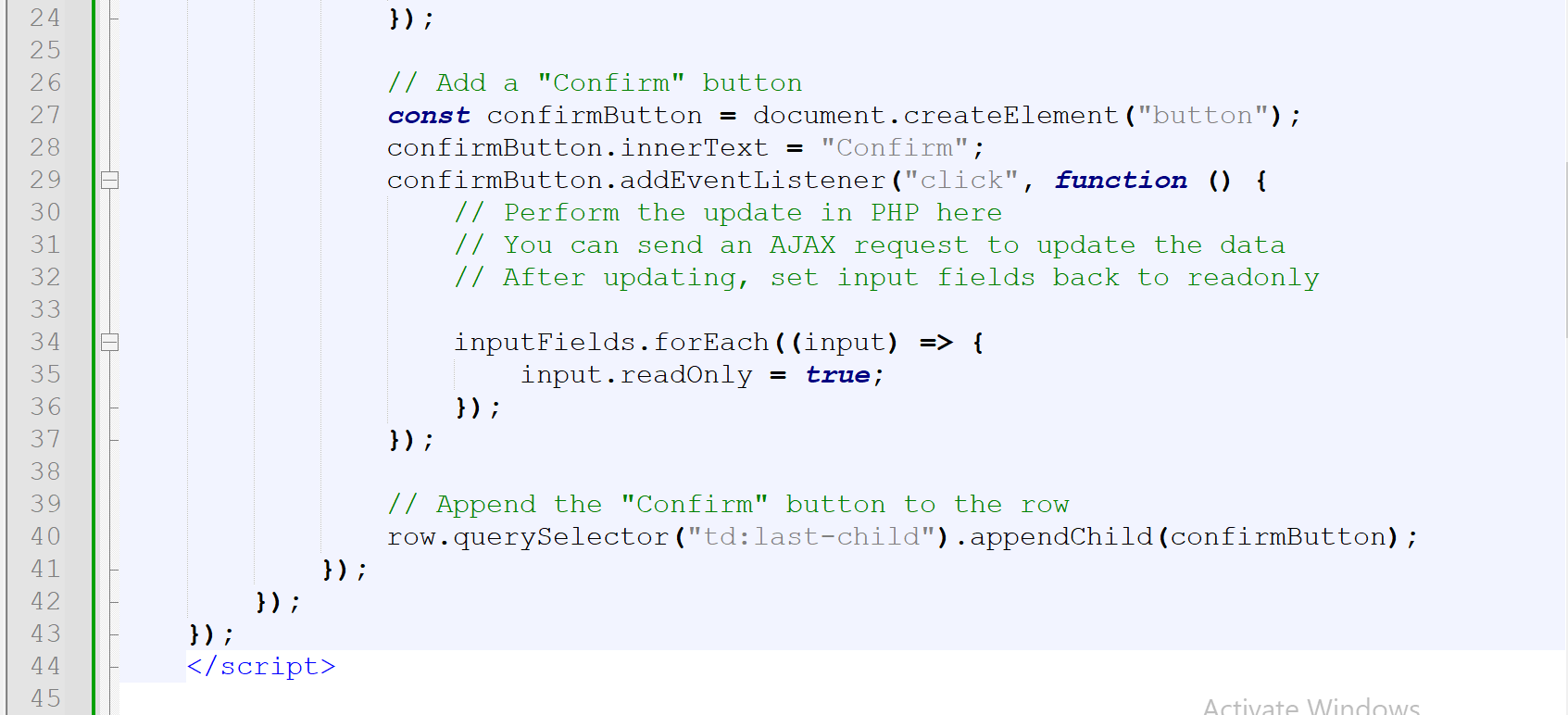
****

**16. Consider the exercise11and add a edit link near delete link e.g. Clicking up on edit button a particular row should be open in editing mode e.g. on the Particular row there should be filled text box with data and on the option column there should be a confirm button clicking upon it arrow should be updated.**

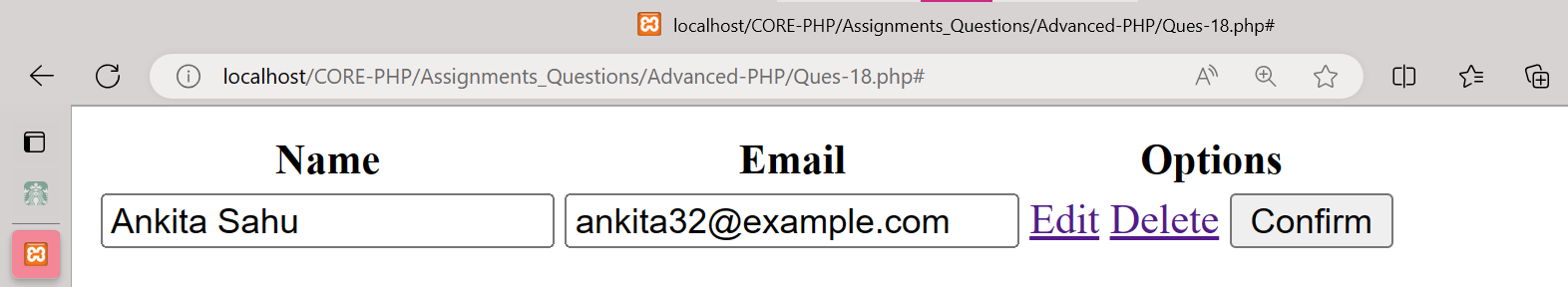
**Ans:-**

**Code:**

****

****

**Output:**

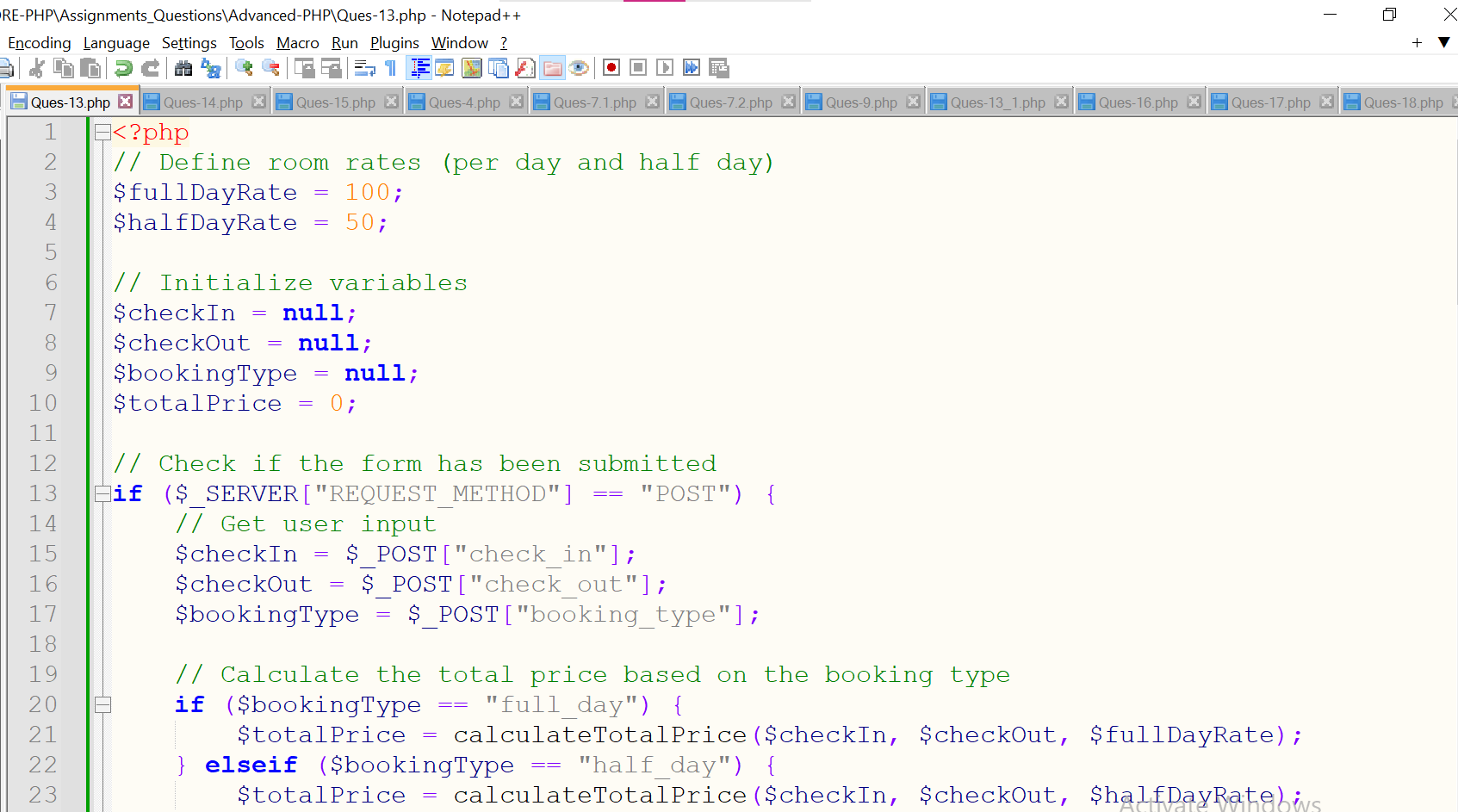
****

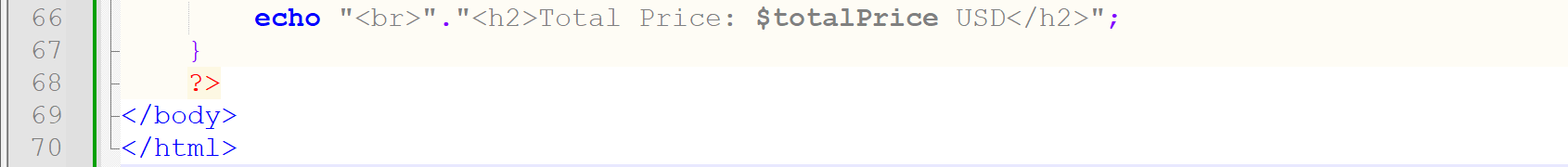
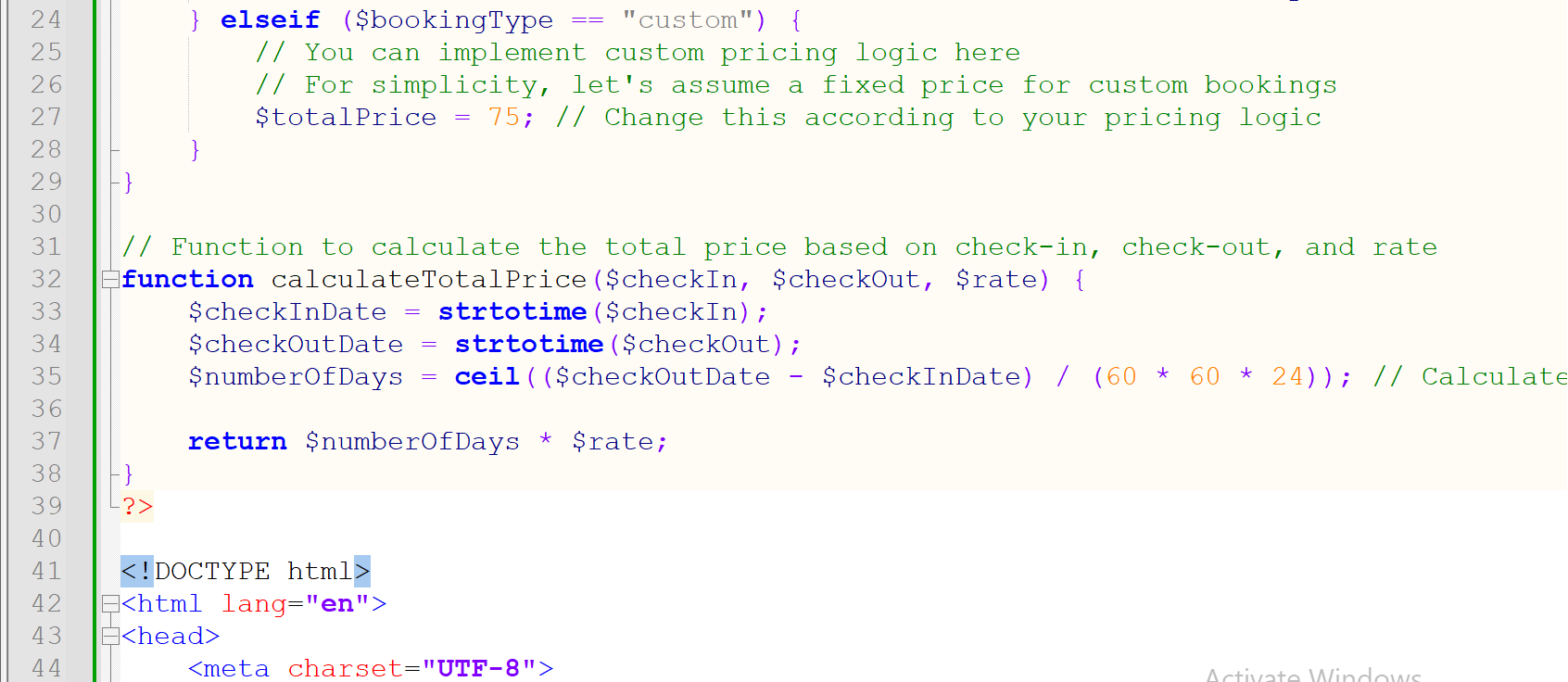
**17. Create Hotel Room Booking System User can book room by 3 ways:**

* **Full day**
* **Half day**
* **Custom**

**Ans:-**

**Code:**

****

****

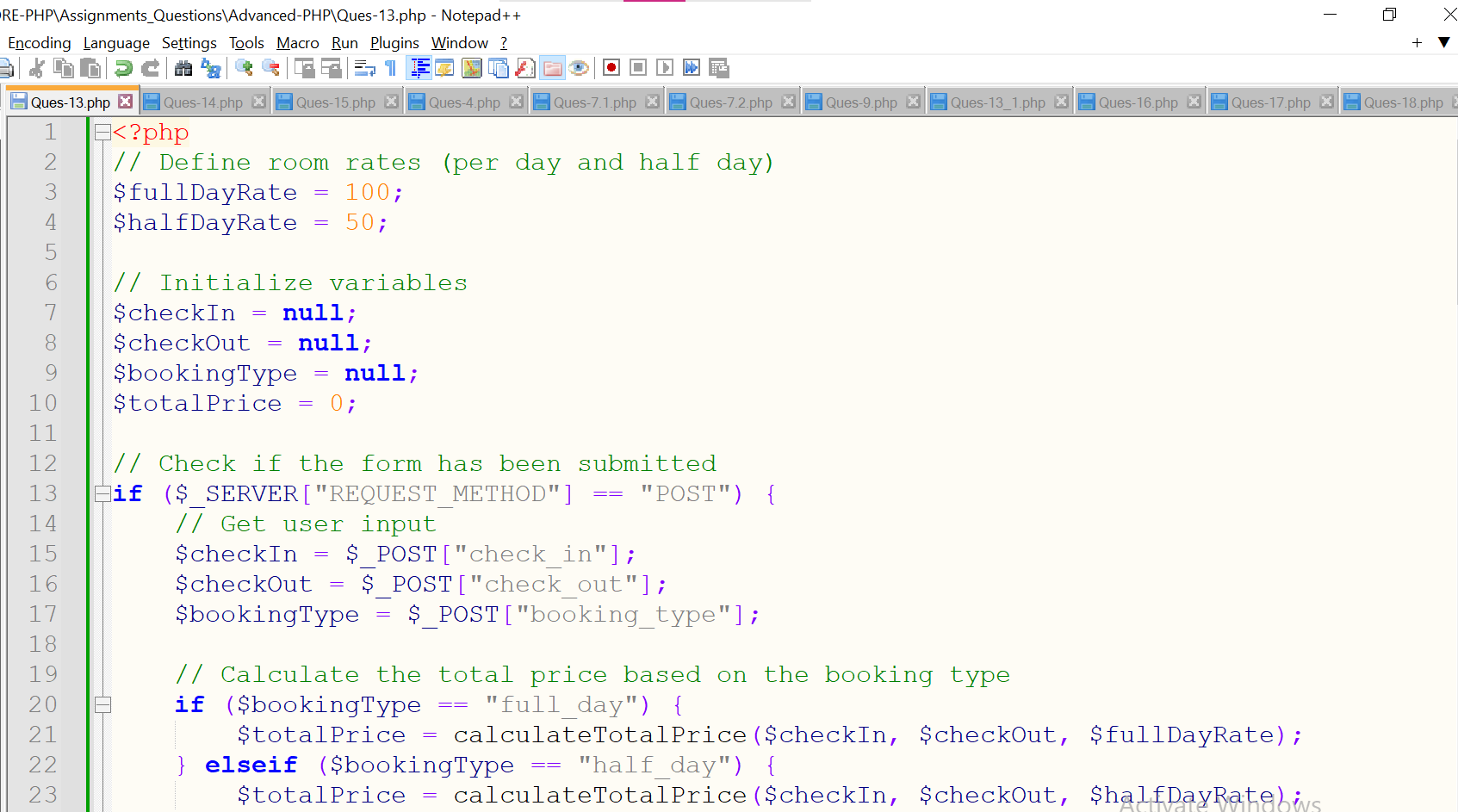
**Output:**

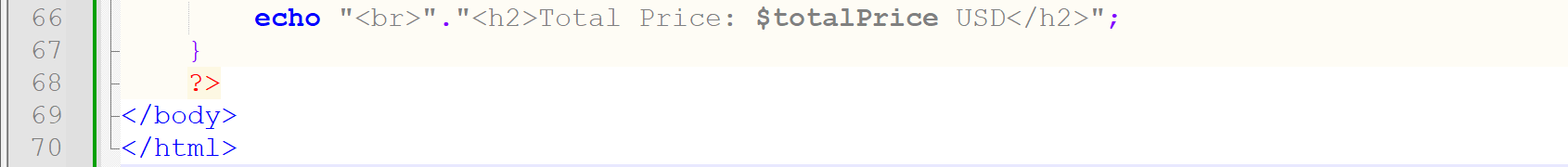
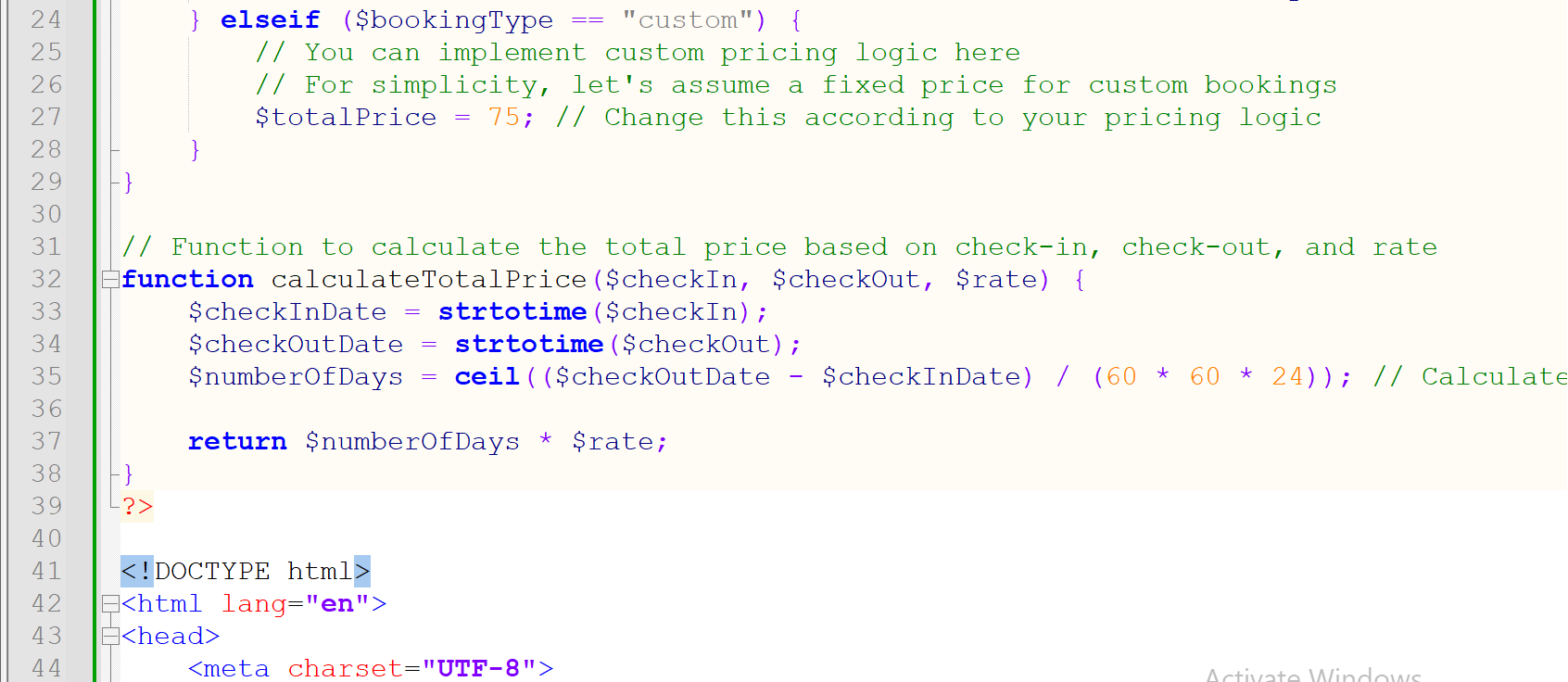
****

**18. If user select for the full day than user only have selection for the checking-checkout date.**

**Ans:-**

**Code:**

****

****

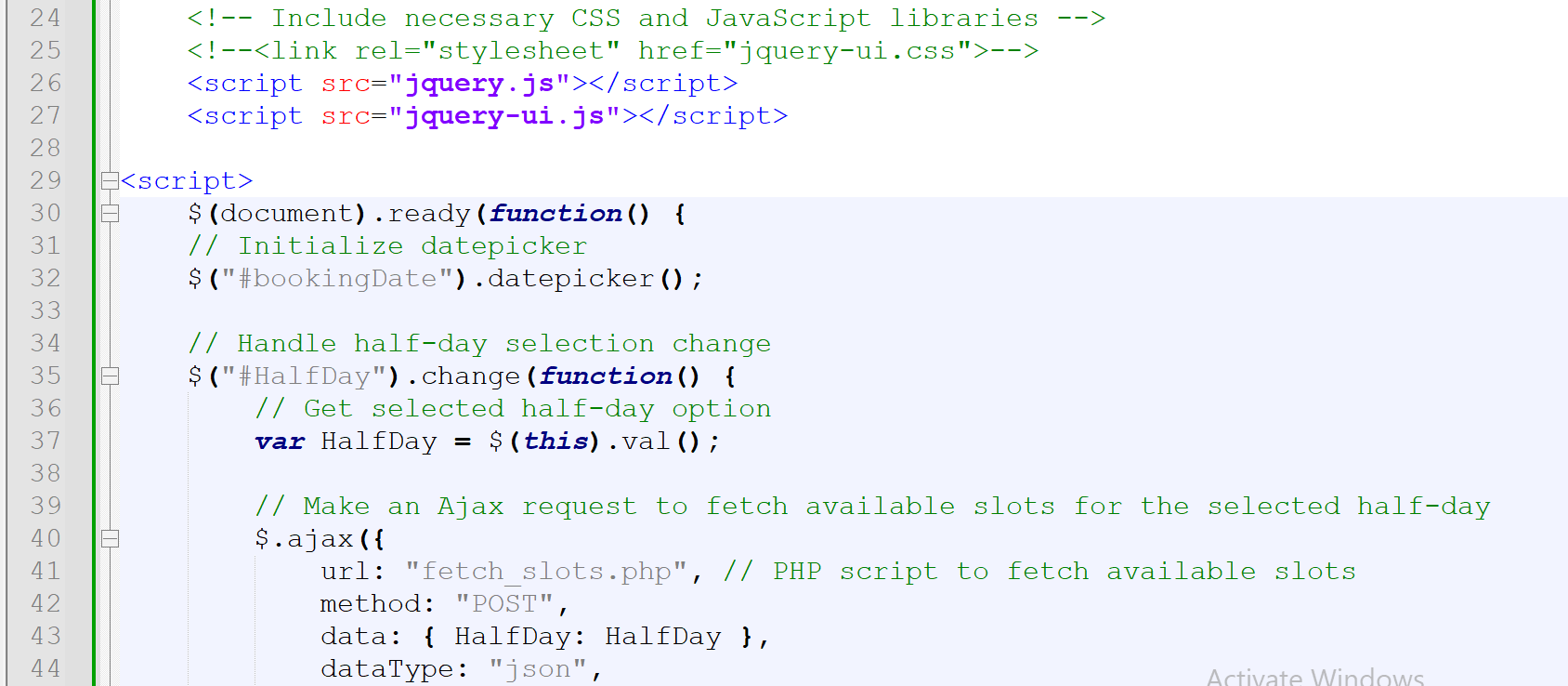
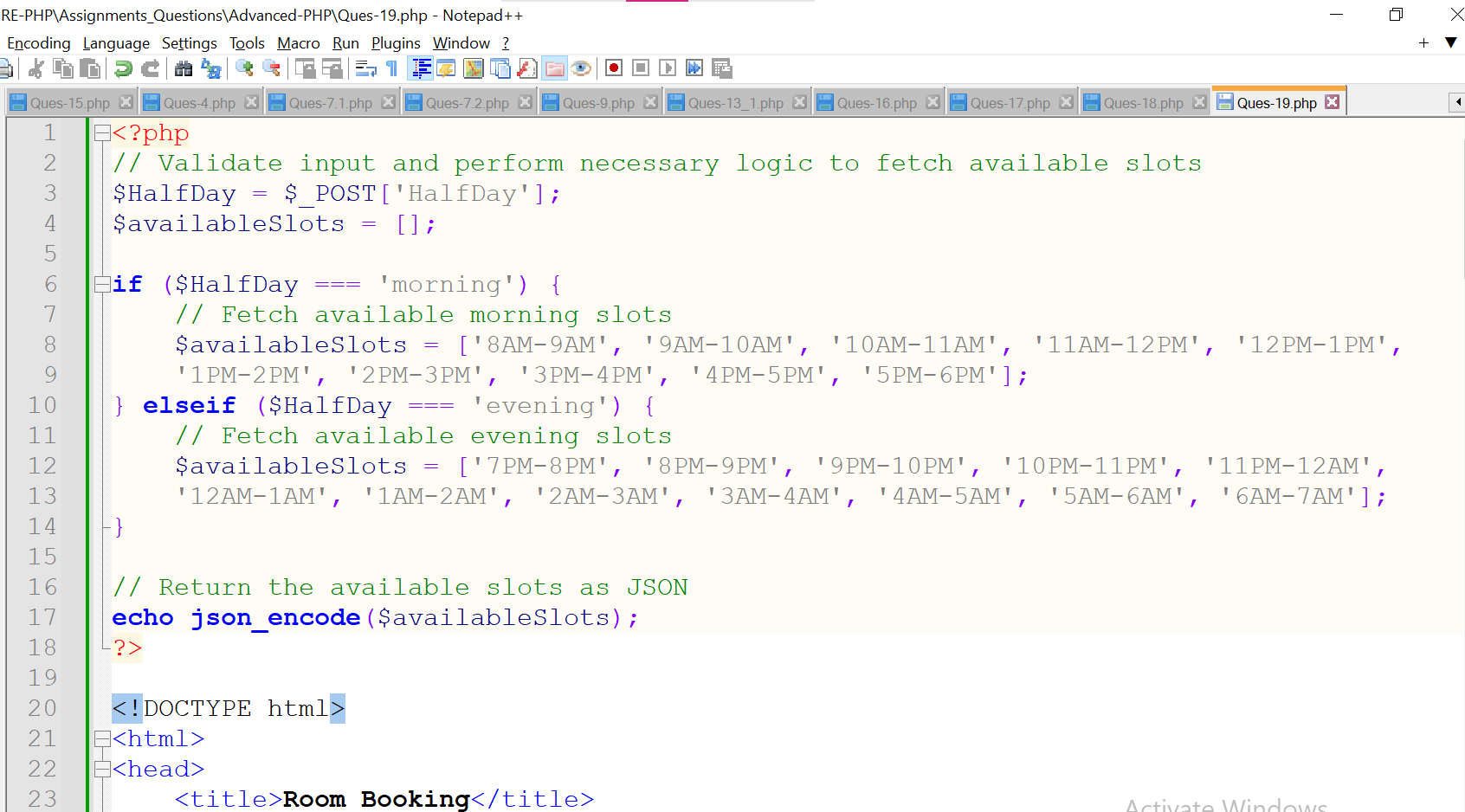
**Output:**

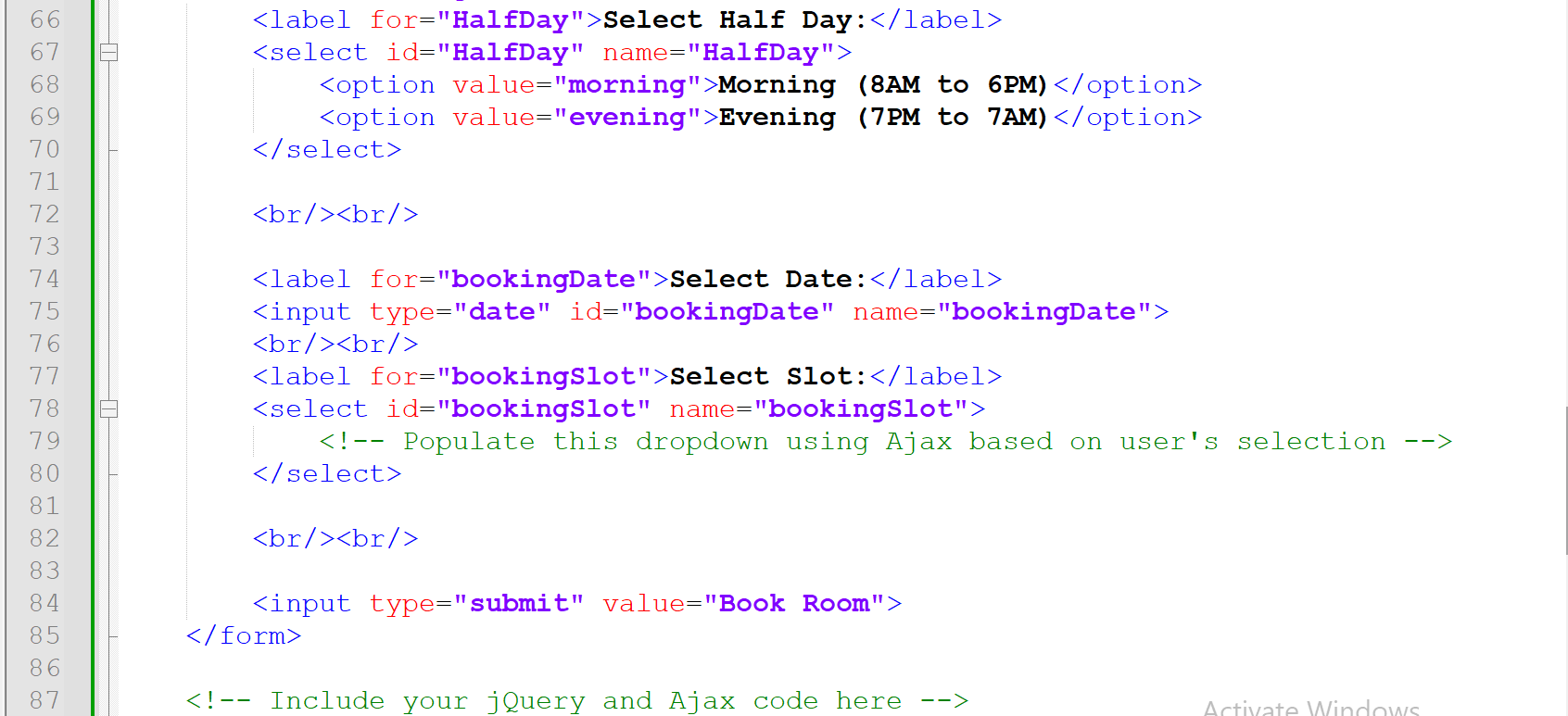
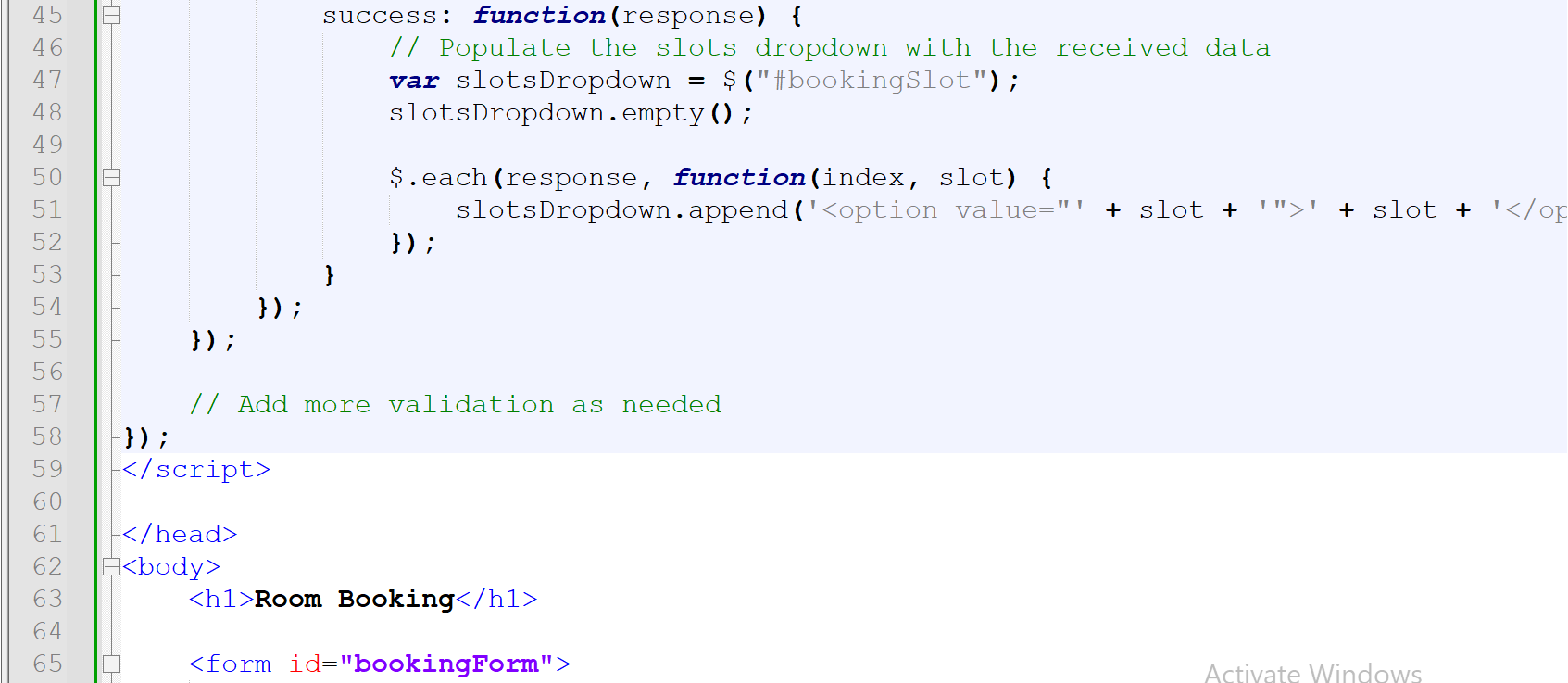
****

**19. If user select Half day than user have option of date and slot option(like user want to book room for first half – Morning (8AM to 6PM) if user select for second halfit‟s for evening (7PM to Morning 7AM)). Do proper validation like if user can book only available slot. (have touse jQuery -> Ajax, validation, Json passing).**

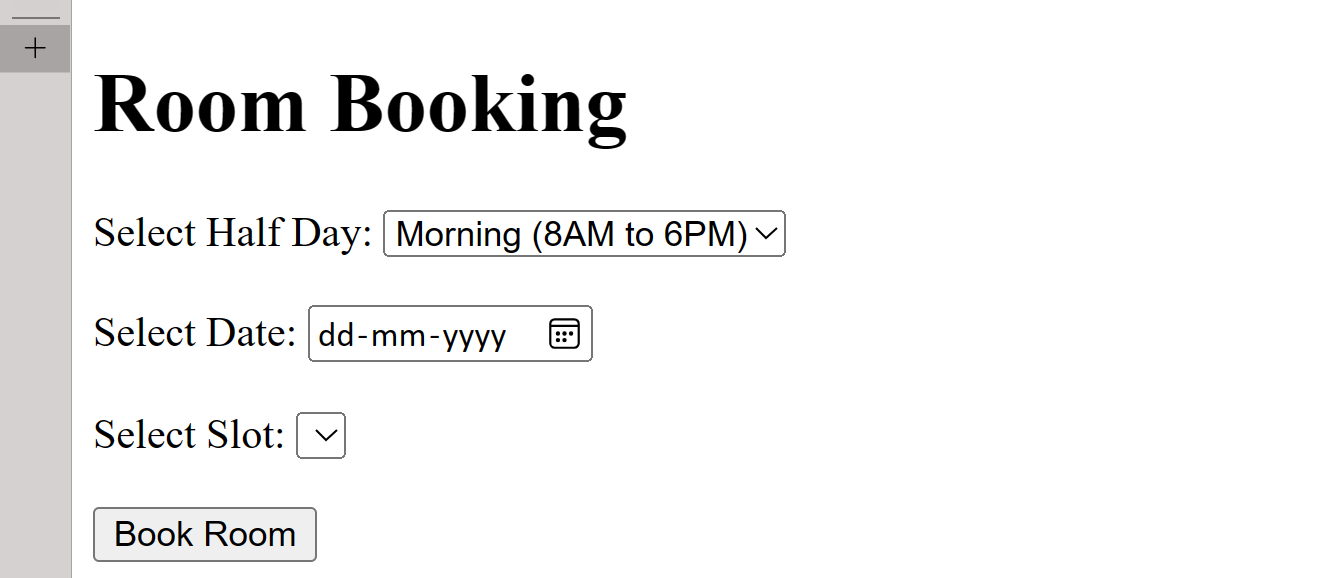
**Ans:-**

**Code:**

****

****

**Output:**

****

**XOXOXOXOXOXOXOXOXOXOXO**