MENEDŻERSKA AKADEMIA NAUK STOSOWANYCH W WARSZAWIE 51 DPH COMPUTER ENGINEERING

****

# PREPARED BY

## ENFAL SEVİNÇ, 77789

PROGRAMMING IN SCRIPTING LANGUAGES

# SUPERVISOR

## KUMAR NALINAKSH

2023 POLAND

[Question 1 4](#_Toc134817572)

[Question 2 4](#_Toc134817573)

[Question 3 4](#_Toc134817574)

[Question 4 5](#_Toc134817575)

[Question 5 5](#_Toc134817576)

[Question 6 5](#_Toc134817577)

[Question 7 5](#_Toc134817578)

[Question 8 5](#_Toc134817579)

[Question 9 6](#_Toc134817580)

[Question 10 6](#_Toc134817581)

[Question 11 6](#_Toc134817582)

[Question 12 6](#_Toc134817583)

[Question 13 6](#_Toc134817584)

[Question 14 7](#_Toc134817585)

[Question 15 7](#_Toc134817586)

[Question 16 7](#_Toc134817587)

[Question 17 7](#_Toc134817588)

[Question 18 8](#_Toc134817589)

[Question 19 8](#_Toc134817590)

[Question 20 8](#_Toc134817591)

[Question 21 8](#_Toc134817592)

[Question 22 9](#_Toc134817593)

[Question 23 9](#_Toc134817594)

[Question 24 9](#_Toc134817595)

[Question 25 9](#_Toc134817596)

[Question 26 9](#_Toc134817597)

[Question 27 10](#_Toc134817598)

[Question 28 10](#_Toc134817599)

[Question 29 10](#_Toc134817600)

[Question 30 10](#_Toc134817601)

[Question 31 11](#_Toc134817602)

[Question 32 11](#_Toc134817603)

[Question 33 11](#_Toc134817604)

[Question 34 11](#_Toc134817605)

[Question 35 12](#_Toc134817606)

[Question 36 12](#_Toc134817607)

[Question 37 12](#_Toc134817608)

[Question 38 12](#_Toc134817609)

[Question 39 13](#_Toc134817610)

[Question 40 13](#_Toc134817611)

[Question 41 13](#_Toc134817612)

[Question 42 13](#_Toc134817613)

[Question 43 14](#_Toc134817614)

[Question 44 14](#_Toc134817615)

[Question 45 14](#_Toc134817616)

[Question 46 14](#_Toc134817617)

[Question 47 15](#_Toc134817618)

[Question 48 15](#_Toc134817619)

[Question 49](#_Toc134817620) 15

## Question 1

What is the purpose of classes and objects in Python?

Answer

Classes provide a means of bundling data and functionality together. Creating a new class creates a new type of object, allowing new instances of that type to be made. Each class instance can have attributes attached to it for maintaining its state.

## Question 2

How do you define a class in Python?

Answer

A class in Python can be defined using the class keyword. As per the syntax above, a class is defined using the class keyword followed by the class name and : operator after the class name, which allows you to continue in the next indented line to define class members.

## Question 3

How do you create an object of a class in Python?

Answer

Define the class: Begin by defining the class using the class keyword followed by the class name. Inside the class, you can define attributes (data) and methods (functions). Instantiate the object: To create an object from the class, you need to call the class as if it were a function. This process is known as instantiation. When you instantiate an object, memory is allocated to store its data.

## 

## Question 4

What is data abstraction and hiding through classes?

Answer

Abstraction is the concept of hiding internal implementation details and expose only the necessary functionalities or apis to the end users. Data Hiding is the concept of restricting access to class and instance variables to improve the security aspect of the applications

## Question 5

How do you implement data hiding in Python classes?

Answer

Data hiding in Python is done by using a double underscore before (prefix) the attribute name. This makes the attribute private/ inaccessible and hides them from users.

## Question 6

What is a class method in Python?

Answer

A class method is a method that is bound to a class rather than its object. It doesn't require creation of a class instance, much like staticmethod. The difference between a static method and a class method is: Static method knows nothing about the class and just deals with the parameters.

## Question 7

What is the difference between a class method and an instance method?

Answer

## Instance methods need a class instance and can access the instance through self . Class methods don't need a class instance. They can't access the instance ( self ) but they have access to the class itself via cls.

## Question 8

What is the purpose of the init() method in Python classes?

Answer

## The \_\_init\_\_ method lets the class initialize the object's attributes and serves no other purpose.

## Question 9

### How do you define a private data member in Python classes?

Answer

In Python, a private member can be defined by using a prefix \_\_ (double underscore). So, in the private modifier's case, we cannot access the attribute.

## Question 10

How do you define a private method in Python classes?

Answer

In python programming, there are no private methods that cannot be accessed except inside the class. To define the private method, you have to prefix the member name with a double underscore(\_\_).

## Question 11

How do you call a class method from another class method in Python?

Answer

To call a class method from another class method in Python, you can use the @classmethod decorator along with the class name. The @classmethod decorator is used to define a class method.

## Question 12

What are thebuilt-in functions in Python for checking class attributes?

Answer

hasattr(object, attribute) getattr(object, attribute[, default]) setattr(object, attribute, value) delattr(object, attribute)

## Question 13

### How do you get the value of a class attribute in Python?

Answer

Use dot notation or getattr() function to get the value of a class attribute. Use dot notation or setattr() function to set the value of a class attribute. Python is a dynamic language. Therefore, you can assign a class variable to a class at runtime

## Question 14

How do you set the value of a class attribute in Python?

Answer

Use dot notation or setattr() function to set the value of a class attribute. Python is a dynamic language. Therefore, you can assign a class variable to a class at runtime. Python stores class variables in the \_\_dict\_\_ attribute.

## Question 15

How do you delete a class attribute in Python?

Answer

Python delattr() function is used to delete an attribute from a class. It takes two parameters first is an object of the class and second is an attribute which we want to delete. After deleting the attribute, it no longer available in the class and throws an error if try to call it using the class object.

## Question 16

What are built-in class attributesin Python?

Answer

\_\_name\_\_:This attribute holds the name of the class as a string.

\_\_module\_\_:This attribute holds the name of the module in which the class is defined.

\_\_bases\_\_:It lists the classes from which the current class inherits.

\_\_dict\_\_:It stores the class's attributes (both data and methods) as key-value pairs.

## Question 17

### What is the purpose of the del() method in Python classes?

Answer

The del keyword is used to delete objects. In Python everything is an object, so the del keyword can also be used to delete variables, lists, or parts of a list etc.

## Question 18

How do you override a method in Python classes?

Answer

Overriding a method in the same class is not allowed. So, you need to do that in the child class by implementing the Inheritance concept. If you want to override the Parent Class method, create a function in the Child with the same name and number of parameters. This is called function overriding in Python.

## Question 19

What is the difference between a public and a private data member in Python classes?

Answer

The main difference between public and private variables is their visibility. Public variables can be accessed by any code that has a reference to the object, while private variables can only be accessed within the class in which they are defined.

## Question 20

What is the difference between a public and a private method in Python classes?

Answer

There are three types of access modifiers in Python: public, private, and protected. Variables with the public access modifiers can be accessed anywhere inside or outside the class, the private variables can only be accessed inside the class, while protected variables can be accessed within the same package.

## Question 21

How do you implement inheritance in Python classes?

Answer

Define the parent class: Start by defining the parent class with its attributes and methods.Define the child class: Create the child class and specify the parent class inside parentheses after the child class name.Customize the child class: Optionally, you can add additional attributes and methods to the child class or override the parent class's attributes and methods.

## Question 22

### What is the purpose of the super() function in Python classes?

Answer

The super() function is used to give access to methods and properties of a parent or sibling class. The super() function returns an object that represents the parent class.

## Question 23

What is the difference between a parent class and a child class in Python?

Answer

Parent class is the class being inherited from, also called base class. Child class is the class that inherits from another class, also called derived class.

## Question 24

How do you define an abstract class inPython?

Answer

Import the ABC class and the abstractmethod decorator from the abc module.Create a class and inherit from ABC as the base class.Use the @abstractmethod decorator to decorate the methods that you want to make abstract. These methods must not have an implementation in the abstract class.

## Question 25

How do you implement abstract methods in Python classes?

Answer

To define an abstract method we use the @abstractmethod decorator of the abc module. It tells Python that the declared method is abstract and should be overridden in the child classes. We just need to put this decorator over any function we want to make abstract, and the abc module takes care of the rest.

## Question 26

What is polymorphism in Python classes?

Answer

The word "polymorphism" means "many forms", and in programming it refers to methods/functions/operators with the same name that can be executed on many objects or classes.

## Question 27

How do you implement polymorphism in Python classes?

Answer

Define a common superclass: Create a superclass with a method that will be overridden by the subclasses.Define subclasses: Create one or more subclasses that inherit from the superclass and provide their own implementation of the overridden method.Use the common method: Use the common method on objects of different classes, treating them as instances of the superclass.

## Question 28

What is encapsulation in Python classes?

Answer

Encapsulation is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as a single unit. In encapsulation, the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class.

## Question 29

How do you implement encapsulation in Python classes?

Answer

Encapsulation can be achieved by declaring the data members and methods of a class either as private or protected. But In Python, we don't have direct access modifiers like public, private, and protected. We can achieve this by using single underscore and double underscores

## Question 30

### What is a decorator in Python classes?

Answer

A decorator in Python is any callable Python object that is used to modify a function or a class. A reference to a function "func" or a class "C" is passed to a decorator and the decorator returns a modified function or class.

## Question 31

What is the difference between a static method and a class method in Python?

Answer

Class methods can access and modify class-level attributes. They have access to the class object and can modify class variables or create new instances of the class. Static methods, on the other hand, do not have access to the class object and cannot modify any class-level attributes.

## Question 32

How do you define a static method in Python classes?

Answer

To make a method a static method, add @staticmethod decorator before the method definition. The @staticmethod decorator is a built-in function decorator in Python to declare a method as a static method. It is an expression that gets evaluated after our function is defined.

## Question 33

What is the purpose of the @classmethod decorator in Python classes?

Answer

## In Python, the @classmethod decorator is used to declare a method in the class as a class method that can be called using ClassName. MethodName() . The class method can also be called using an object of the class.

## Question 34

How do you implement multiple inheritance in Python classes?

Answer

Define the base classes: Create the base classes, each with their own attributes and methods.Define the derived class: Create the derived class and specify the base classes inside parentheses after the derived class name, separated by commas.Access attributes and methods: You can access attributes and methods from all the base classes as well as define additional attributes and methods in the derived class.

## Question 35

What is the purpose of the str() method in Python classes?

Answer

The python \_\_str\_\_ method returns the object representation in a string format. This method is supposed to return a human-readable format which is used to display some information about the object.

## Question 36

How do you define a property in Python classes?

Answer

In Python, a property in the class can be defined using the property() function. The property() method in Python provides an interface to instance attributes. It encapsulates instance attributes and provides a property.

## Question 37

What is the purpose of the @property decorator in Python classes?

Answer

The @property is a built-in decorator for the property() function in Python. It is used to give "special" functionality to certain methods to make them act as getters, setters, or deleters when we define properties in a class.

## Question 38

How do you implement a setter method in Python classes?

Answer

In Python, you can implement a setter method to control the assignment of values to attributes of a class. A setter method allows you to define custom logic and validations when setting the value of an attribute. To implement a setter method in a Python class, you can use the @property decorator and the corresponding setter decorator, @<attribute>.setter.

## 

## Question 39

How do you implement a deleter method in Python classes?

Answer

In Python, you can implement a deleter method to control the deletion or removal of an attribute from a class. A deleter method allows you to define custom logic and actions when an attribute is deleted. To implement a deleter method in a Python class, you can use the @property decorator and the corresponding deleter decorator, @<attribute>.deleter.

## Question 40

What is a metaclass in Python classes?

Answer

A metaclass in Python is a class of a class that defines how a class behaves. A class is itself an instance of a metaclass. A class in Python defines how the instance of the class will behave. In order to understand metaclasses well, one needs to have prior experience working with Python classes.

## Question 41

How do you implement operator overloading in Python classes?

Answer

Choose the operator: Determine which operator you want to overload for your class.Implement the corresponding special method: Every operator has a corresponding special method that you can define in your class. For example, to overload the addition operator (+), you can implement the \_\_add\_\_ method.Define the desired behavior: Inside the special method, define the behavior that should occur when the operator is used on objects of your class.

## Question 42

### What is the difference between a class attribute and an instance attribute in Python?

Answer

In summary, class attributes remain the same for every object and are defined outside the \_\_init\_\_() function. Instance attributes are somewhat dynamic because they can have different values in each object. Instance attributes are defined in the \_\_init\_\_() function.

## Question 43

How do you define a class attribute in Python classes?

Answer

A class attribute is shared by all instances of the class. To define a class attribute, you place it outside of the \_\_init\_\_() method. Use class\_name. class\_attribute or object\_name.

## Question 44

What is the purpose of the new() method in Python classes?

Answer

When you create an instance of a class, Python first calls the \_\_new\_\_() method to create the object and then calls the \_\_init\_\_() method to initialize the object's attributes. The first argument of the \_\_new\_\_ method is the class of the new object that you want to create.

## Question 45

What is a descriptor in Python classes?

Answer

Descriptors are a powerful, general purpose protocol. They are the mechanism behind properties, methods, static methods, class methods, and super() . They are used throughout Python itself. Descriptors simplify the underlying C code and offer a flexible set of new tools for everyday Python programs.

## Question 46

How do you implement a descriptor in Python classes?

Answer

Define a descriptor class: Create a separate class that will serve as the descriptor. This class should implement one or more of the descriptor methods: \_\_get\_\_, \_\_set\_\_, or \_\_delete\_\_.Associate the descriptor with a class attribute: In the class where you want to use the descriptor, create an instance of the descriptor class as a class attribute.Use the descriptor attribute: Access, assign, or delete the descriptor attribute through instances of the class.

## Question 47

What is the difference between an abstract class and an interface in Python?

Answer

An abstract class can contain both abstract and non-abstract methods, whereas an Interface can have only abstract methods. Abstract classes are extended, while Interfaces are implemented.

## Question 48

How do you implement an interface in Python classes?

Answer

## Question 49

### What is a mixin in Python classes?

Answer

A mixin is a class that provides methods to other classes, but it's not considered a base class itself. 00:18 This special class is going to expose some methods that the derived class can utilize—methods that will essentially be mixed in to the derived class.

## Question 50 How do you implement a mixin in Python classes?

Answer

## Inherit from the mixin class: In the classes where you want to incorporate the mixin behavior, include the mixin class as one of the base classes in the class definition.Use the mixin functionality: Access and use the methods or attributes defined in the mixin class, along with those from other inherited base classes.