1. Use to initialize object.
2. Constructor is a block (like methods) having the same name as the class.
3. Does not have any return type not even void.
4. The only modifiers applicable to constructors are public, private, default and protected (static, final, synchronized etc. cannot be used with constructor’s).
5. It executes automatically when object is created.

Ways to call constructors 🡪

1. Test t = new Test()
2. New Test() (No need to create reference)

The constructor is invoked automatically when an object is created. It initializes the new object (sets up its initial state) but does not create the object itself.

The new keyword (in languages like Java or C++) is responsible for allocating memory and creating the object.

Default Constructor (No argument Constructor)

1. It is created by Compiler not JVM.
2. It only created when no other constructors are present.

No argument Constructor (User defined)

Parameterized Constructor

Q1.

A computer screen with colorful text

AI-generated content may be incorrect.

Will it run?

Ans. No this will show error as default or no argument constructor is not present.

Q2. What is a default constructor?

Ans: A default constructor is a constructor that takes no arguments and is provided automatically by the compiler if no other constructors are defined. If you define any constructor, the default constructor is no longer provided unless explicitly defined.

Q3. What is constructor overloading?

Ans: Constructor overloading is a concept where multiple constructors are defined in a class with the same name (the class name) but different parameter lists. This allows you to create objects in different ways, providing flexibility in initialization.

Q4. What is constructor chaining?

Ans Constructor chaining is when one constructor calls another constructor in the same class or superclass. This can be done using this() for the same class or super() for the parent class. It is often used to reduce redundancy in code.

A computer screen shot of text

AI-generated content may be incorrect.

Q5. Can a constructor be inherited?

Ans: No, constructors cannot be inherited. While you can inherit the fields and methods from a parent class, constructors are not inherited by subclasses. However, a subclass can call a parent class's constructor using the super() keyword.

Q6. Can a constructor be private?

Ans: Yes, a constructor can be private. This is commonly used in design patterns like the Singleton pattern, where you want to prevent the creation of objects from outside the class.

A screen shot of a computer program

AI-generated content may be incorrect.

Q7. What is the difference between this() and super() in constructors?

Ans:

this() is used to call another constructor in the same class.

super() is used to call a constructor from the parent class.

Both this() and super() must be the first statement in the constructor.

Q8. What happens if we don’t define a constructor in a class?

Ans: If no constructor is explicitly defined, the compiler provides a default constructor that takes no parameters and initializes instance variables with default values (e.g., 0, null, false).

Q9. If a constructor is private can you create objects of that class?

Ans: If the constructor is private, you won’t be able to create an object outside the class. But inside the class you can create object.