# SNS COLLEGE OF TECHNOLOGY

*(An Autonomous Institution)*
*Approved by AICTE, New Delhi, affiliated to Anna University, Chennai Accredited by NAAC-UGC with 'A++' Grade (Cycle III) &*
*Accredited by NBA*
COIMBATORE-641 035, TAMIL NADU

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

Course Code & Name : **23CSB201 – OBJECT ORIENTED PROGRAMMING**

Course Faculty : **Ms. S. SANGEETHA, AP-CSD**

**Puzzles / In Class Activities**

**Topics Covered: Unit – II**

## Constructor

What is true about private constructor?
a) Private constructor ensures only one instance of a class exist at any point of time
b) Private constructor ensures multiple instances of a class exist at any point of time
c) Private constructor eases the instantiation of a class
d) Private constructor allows creating objects in other classes
Answer: a
Explanation: Object of private constructor can only be created within class. Private constructor is used in singleton pattern.

2. What would be the behaviour if this() and super() used in a method?
a) Runtime error
b) Throws exception
c) compile time error
d) Runs successfully
Answer: c
Explanation: this() and super() cannot be used in a method. This throws compile time error..

3. What is false about constructor?
a) Constructors cannot be synchronized in Java
b) Java does not provide default copy constructor
c) Constructor can have a return type
d) "this" and "super" can be used in a constructor

Answer: c
Explanation: The constructor cannot have a return type. It should create and return new objects. Hence it would give a compilation error.

4. What could be true about Class.getInstance() considering that it defines the getInstance() method?
a) Class.getInstance calls the constructor
b) Class.getInstance is same as new operator
c) Class.getInstance needs to have matching constructor
d) Class.getInstance creates object if class does not have any constructor
Answer: d
Explanation: The method getInstance() can be used to create an object if the class doesn't have any constructor.

5. What is true about constructor?
a) It can contain return type
b) It can take any number of parameters
c) It can have any non access modifiers
d) Constructor cannot throw an exception
Answer: b
Explanation: Constructor returns a new object with variables defined as in the class. Instance variables are newly created and only one copy of static variables are created.

6. Abstract class cannot have a constructor.
a) True
b) False
Answer: b
Explanation: No instance can be created of abstract class. Only pointer can hold instance of object.

7. What is true about protected constructor?
a) Protected constructor can be called directly
b) Protected constructor can only be called using super()
c) Protected constructor can be used outside package
d) protected constructor can be instantiated even if child is in a different package
Answer: b
Explanation: Protected access modifier means that constructor can be accessed by child classes of the parent class and classes in the same package.

8. What is not the use of "this" keyword in Java?
a) Passing itself to another method
b) Calling another constructor in constructor chaining
c) Referring to the instance variable when local variable has the same name
d) Passing itself to method of the same class
Answer: d
Explanation: "this" is an important keyword in java. It helps to distinguish between local variable and variables passed in the method as parameters.

9. What would be the behaviour if one parameterized constructor is explicitly defined, but no default constructor is provided?
a) Compilation error
b) Compilation succeeds
c) Runtime error
d) Compilation succeeds but at the time of creating object using default constructor, it throws compilation error
Answer: d
Explanation: The class compiles successfully. But the object creation of that class gives a compilation error.

10. What would be behaviour if the constructor has a return type?
a) Compilation error
b) Runtime error
c) Compilation and runs successfully
d) Only String return type is allowed
Answer: a
Explanation: The constructor cannot have a return type. It should create and return new object. Hence it would give compilation error.

11. What will be the output of this program?

```
class Demo {
   int num;
   Demo() {
      this(100);
      System.out.println("Default Constructor");     }
   Demo(int n) {
      num = n;
      System.out.println("Parameterized Constructor");     }
}
public class Main {
   public static void main(String[] args) {
      Demo obj = new Demo();
   }
}
```

A Default Constructor
B Parameterized Constructor
C Parameterized Constructor
   Default Constructor
D Compilation Error
Answer: C
Explanation : The this(100); statement calls the parameterized constructor first, followed by the execution of the default constructor.

12. What will be the output of this program?

```
class A {
   private A() {
      System.out.println("Private Constructor");
   }
   public static void createInstance() {
      new A();
   }
}
public class Main {
   public static void main(String[] args) {
      A.createInstance();
   }
}
```

A Compilation Error
B Private Constructor
C Runtime Error
D No Output
Answer: A
Explanation:  A private constructor can still be called inside the class itself using a static method.

13. Can a constructor be final in Java?
A Yes, if you override it.
B No, because constructors are not inherited.
C Yes, but only in abstract classes.
D Yes, but only for utility classes.
Answer: B
Explanation: A constructor cannot be final because it is never inherited, and final is meant to prevent overriding.

14. How can one constructor call another constructor within the same class?
A Using super()
B Using this()
C Using new()
D Constructors cannot call each other
Answer: B
Explanation: In Java, this() is used to call another constructor of the same class, allowing constructor chaining.

15. Can a Java constructor be private?
A No, it must be public.
B Yes, but the class cannot be instantiated outside the class.
C Yes, and it can be accessed from anywhere.
D No, Java does not allow private constructors.
Answer: B
Explanation: A private constructor is used in Singleton design patterns or utility classes to prevent object creation outside the class.

16. Which method is used to explicitly request garbage collection in Java?
a) System.gc()
b) Object.releaseMemory()
c) GarbageCollector.run()
d) MemoryManager.cleanup()
Answer: a

17. Which of these triggers Garbage Collection?
a) delete keyword
b) free() method
c) JVM automatically
d) finalize() manually
Answer: c