

# 26 Lecture - CS304

## Important Mcqs

### 1. What is base initialization in object-oriented programming?

- a. A mechanism to initialize the derived class data members before the base class constructor is called
- b. A mechanism to initialize the base class data members before the derived class constructor is called
- c. A mechanism to initialize both the base class and derived class data members together
- d. A mechanism to dynamically allocate memory for the base class and derived class data members

**Answer: b. A mechanism to initialize the base class data members before the derived class constructor is called**

### Why is base initialization important in C++?

- a. It ensures that the derived class data members are initialized properly
- b. It avoids unnecessary default constructor calls
- c. It allows the programmer to explicitly call the constructor of the base class
- d. All of the above

**Answer: d. All of the above**

### Where is the base initialization list typically located in a C++ constructor?

- a. At the beginning of the constructor body
- b. After the constructor body
- c. Before the constructor declaration
- d. None of the above

**Answer: a. At the beginning of the constructor body**

### Which of the following is an advantage of using base initialization?

- a. It can improve performance by avoiding unnecessary default constructor calls
- b. It can make code more readable and maintainable
- c. It can ensure that base class data members are initialized properly
- d. All of the above

**Answer: d. All of the above**

### What is the syntax for using base initialization in C++?

- a. `baseClassName(argumentList), constructorBody`
- b. `constructorBody, baseClassName(argumentList)`
- c. `baseClassName(argumentList) : constructorBody`
- d. None of the above

**Answer: c. `baseClassName(argumentList) : constructorBody`**

### When should base initialization be used in C++?

- a. When the base class has a parameterized constructor
- b. When the derived class needs to initialize data members that are dependent on the values of the base class data members

- c. When the base class has const data members that cannot be initialized in the derived class constructor
- d. All of the above

**Answer: d. All of the above**

**Which of the following is an example of base initialization in C++?**

- a. A derived class constructor that initializes the base class data members using default values
- b. A derived class constructor that initializes the base class data members using constructor arguments
- c. A derived class constructor that initializes the derived class data members using constructor arguments
- d. None of the above

**Answer: b. A derived class constructor that initializes the base class data members using constructor arguments**

**Can base initialization be used to initialize data members of both the base class and derived class?**

- a. Yes, it can be used to initialize both the base class and derived class data members
- b. No, it can only be used to initialize the base class data members
- c. It depends on the constructor arguments passed in
- d. None of the above

**Answer: b. No, it can only be used to initialize the base class data members**

**Which of the following is an example of a scenario where base initialization is useful?**

- a. When a derived class needs to allocate memory for the base class data members
- b. When a derived class needs to call the default constructor of the base class
- c. When a derived class needs to initialize const data members in the base class
- d. None of the above

**Answer: c. When a derived class needs to initialize const data members in the base class**

**What is the difference between base initialization and default initialization in C++?**

- a. Base initialization initializes the base class data members, while default initialization initializes the derived class data members