

# 17 Lecture - CS304

## Important Mcqs

1. **What is the syntax for overloading the assignment operator in C++?**

- A) operator = ()
- B) operator ()
- C) operator ()
- D) operator += ()

**Answer: A**

**Which of the following is a valid signature for an overloaded assignment operator that takes a reference to the class object as a parameter?**

- A) MyClass operator=(MyClass& obj)
- B) void operator=(const MyClass& obj)
- C) MyClass& operator=(const MyClass& obj)
- D) void operator=(MyClass& obj)

**Answer: C**

**What is the return type of an overloaded assignment operator?**

- A) void
- B) the class type
- C) int
- D) bool

**Answer: B**

**How is the overloaded assignment operator invoked in C++?**

- A) using the = operator
- B) using the copy constructor
- C) using the constructor
- D) using the destructor

**Answer: A**

**What is the purpose of overloading the assignment operator in C++?**

- A) to allow objects of a class to be assigned values using the = operator
- B) to allow objects of a class to be compared using the == operator
- C) to allow objects of a class to be initialized using the constructor
- D) to allow objects of a class to be destroyed using the destructor

**Answer: A**

**Which of the following is true about the copy constructor and the assignment operator?**

- A) they both take a reference to the class object as a parameter
- B) they both return a reference to the class object
- C) they both perform a shallow copy of the object's data members
- D) they both perform a deep copy of the object's data members

**Answer: D**

**What is the difference between the copy constructor and the assignment operator?**

- A) the copy constructor creates a new object, while the assignment operator modifies an

existing object

B) the copy constructor takes a const reference to the object, while the assignment operator takes a non-const reference

C) the copy constructor performs a deep copy of the object, while the assignment operator performs a shallow copy

D) the copy constructor returns a reference to the object, while the assignment operator returns void

**Answer: A**

**How do you avoid issues with self-assignment when overloading the assignment operator?**

A) by checking for self-assignment using the == operator

B) by using a copy constructor to create a new object and then assigning it to the existing object

C) by checking for self-assignment using the this pointer

D) by using a swap function to swap the contents of the object with a temporary object

**Answer: D**

**Which of the following is a common practice when overloading the assignment operator?**

A) returning a copy of the object from the function

B) returning a reference to the object from the function

C) using dynamic memory allocation to perform a deep copy of the object's data members

D) using the default implementation of the operator provided by the compiler

**Answer: C**

**Which of the following is true about the return type of the overloaded assignment operator?**

A) it must be a built-in type such as int or bool

B) it can be any user-defined type

C) it must be the same type as the class being overloaded

D) it can be a different type from the class being overloaded

**Answer: C**