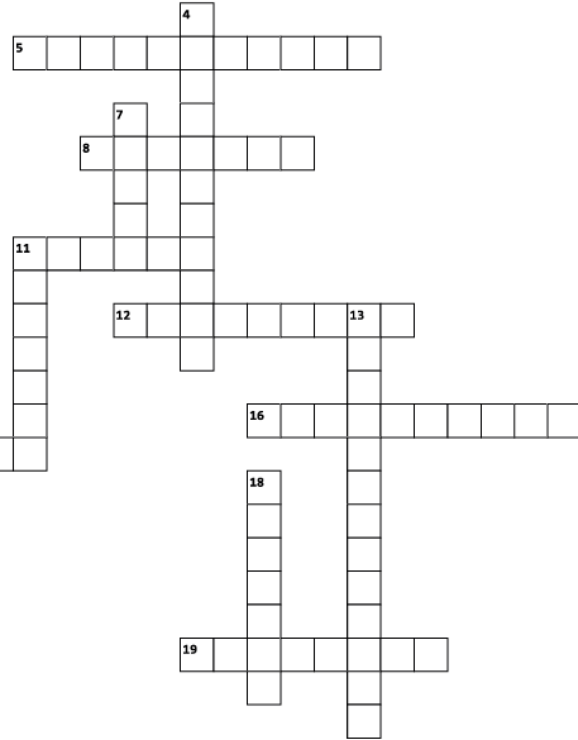
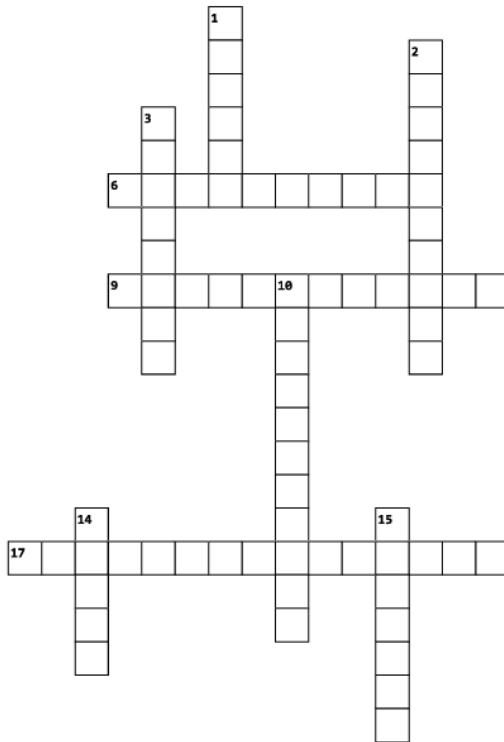


# OOP IA1



## Across

### Shallow Copy

**5.** The copy that duplicates an object's memory address but not the data it points to.

### Destructor

**6.** The function called automatically when an object goes out of scope, used for cleanup.

### virtual

**8.** The keyword that enables a method to be overridden in a derived class.

### Polymorphism

**9.** The OOP principle of using a single interface to represent different underlying forms.

### Public Inheritance

**11.** The mode of inheritance where public members of the base class retain their accessibility.

### Protected Inheritance

**12.** The mode of inheritance where all base class members become protected in the derived class.

### Encapsulation

**16.** The OOP principle of restricting the direct access to an object's data members from outside the class.

### Multiple Inheritance

**17.** The C++ feature that allows a class to be derived from more than one base class.

**19.** The C++11 specifier that ensures a method is correctly overriding a virtual function.

**override**

## Down

**1.** An instance of a class, with its own unique state and behavior. **Object**

**2.** The process of providing a new implementation for an inherited method. **Method Overriding**

**3.** The type of constructor that prevents shallow copies and memory issues in classes with dynamic memory. **Copy Constructor**

**4.** A special member function that is automatically called when an object is created. **Constructor**

**7.** The keyword used to prevent a virtual method from being overridden. **final**

**10.** The OOP concept of defining multiple functions with the same name but different parameters. **Function Overloading**

**11.** The C++ access specifier that restricts a member to be accessed only within its own class. **private**

**13.** The OOP principle of bundling data and methods, hiding the internal state. **Encapsulation**

**14.** The blueprint for creating an object. **Class**

**15.** The type of constructor that has no parameters and is provided automatically if no other constructors are defined. **Default Constructor**

**18.** The term for a class that inherits from another class. **Derived Class**