Short Notes: Object Oriented Programming (OOP) in C++

BASICS OF OOP

- Object-Oriented Programming: A paradigm using classes and objects to design software.
- Class: Blueprint for creating objects (e.g., class Student). Contains data members and member functions.
- Object: Instance of a class. Created at runtime.

CORE OOP CONCEPTS

1. Encapsulation:
   - Binds data and functions into a single unit (class).
   - Uses access specifiers like private, public.
   - Enables data hiding.

2. Abstraction:
   - Hides complex details, shows essential features.
   - Focuses on what an object does, not how.

3. Inheritance:
   - Reuse code from a base class in a derived class.
   - Types:
     - Single
     - Multiple
     - Multilevel
     - Hierarchical
     - Hybrid

4. Polymorphism:

- Same function behaves differently for different objects.
- Types:
  - Compile-time (Static): Function Overloading
  - Runtime (Dynamic): Function Overriding via virtual functions

FUNCTIONS & CONCEPTS

- Function Overloading: Same function name, different parameters.
- Function Overriding: Child class redefines parent class function.
- Virtual Function: Enables runtime polymorphism.
- Pure Virtual Function: Makes a class abstract (used for interfaces).
- Friend Function: Can access private members of a class.
- this Pointer: Refers to the current object instance.

CONSTRUCTORS & DESTRUCTORS

- Constructor: Initializes objects. Types:
  - Default
  - Parameterized
  - Copy Constructor
- Destructor: Cleans up resources. Declared with ~ClassName().

OTHER KEY TOPICS

- Abstract Class: Contains at least one pure virtual function.
- Namespace: Prevents name conflicts in large projects.
- Access Specifiers:
  - private: Accessible within class.
  - public: Accessible anywhere.

- protected: Accessible in derived classes.

ADDITIONAL NOTES

- Operator Overloading: Customizes operators for class objects.

- Virtual Inheritance: Avoids multiple copies in inheritance.

- delete / delete[]: Frees memory allocated with new/new[].