Core OOP Concepts

1. **Classes and Objects**
   - **Definition and creation of classes:** Learn how to define a class, including attributes and methods.
   - **Objects and instances:** Understand how to create objects from classes and use them.
   - **Constructors and destructors:** Study how constructors initialize objects and destructors clean up resources.

2. **Inheritance**
   - **Single and multiple inheritance:** Understand the basics of single inheritance and the complexities of multiple inheritance (if the language supports it).
   - **Base and derived classes:** Learn how derived classes inherit properties and methods from base classes.
   - **Constructor and destructor calls in inheritance:** Understand the order of constructor and destructor calls in an inheritance hierarchy.
   - **Overriding and hiding of methods:** Learn how to override base class methods and the concept of method hiding.

3. **Polymorphism**
   - **Compile-time polymorphism (method overloading, operator overloading):** Understand how to use method overloading and operator overloading to achieve polymorphism.
   - **Run-time polymorphism (virtual functions, pure virtual functions):** Study the use of virtual functions to achieve runtime polymorphism, and the role of pure virtual functions in abstract classes.

4. **Encapsulation**
   - **Private, protected, and public access specifiers:** Learn the different access levels and how they protect data.
   - **Getter and setter methods:** Understand how to use getter and setter methods to control access to private data.
   - **Data hiding:** Study the principle of data hiding to protect object integrity.

5. **Abstraction**
   - **Abstract classes:** Understand the concept of abstract classes and their role in providing a base for derived classes.
   - **Interfaces:** Learn how interfaces can be used to define a contract for classes (relevant in languages like Java).

6. **Dynamic Memory Management**
   - **Pointers and references:** Study the use of pointers and references for dynamic memory management.
   - **Dynamic allocation and deallocation:** Learn how to use new/delete (C++) or malloc/free (C) for dynamic memory.
   - **Smart pointers:** Understand the use of smart pointers (unique_ptr, shared_ptr) in modern C++ for safer memory management.

Advanced OOP Concepts

1. **Templates and Generics**
   - **Class templates:** Learn how to create class templates for generic programming.
   - **Function templates:** Study the use of function templates to create functions that work with any data type.
   - **Generic programming:** Understand the concept of generics in languages like Java.

2. **Exception Handling**
   - **Try, catch, and throw statements:** Learn the syntax and use of try, catch, and throw for exception handling.
   - **Custom exceptions:** Understand how to define and use custom exception classes.

3. **Design Patterns**
   - **Creational patterns:** Study patterns like Singleton, Factory, and Abstract Factory.
   - **Structural patterns:** Learn about Adapter, Composite, and Proxy patterns.
   - **Behavioral patterns:** Understand patterns like Strategy, Observer, and Command.