Theory Question

27\_jan\_se NAME: Vrinda Mavadhiya

## 1. What are the key differences between Procedural Programming and ObjectOriented Programming (OOP)?

**Key Differences Between Procedural Programming (POP) and Object-Oriented Programming (OOP):**

| **Feature** | **Procedural Programming (POP)** | **Object-Oriented Programming (OOP)** |
| --- | --- | --- |
| **Approach** | Top-down approach | Bottom-up approach |
| **Focus** | Focuses on functions or procedures | Focuses on objects and classes |
| **Data Access** | Data is global and accessible by any function | Data is encapsulated within objects |
| **Modularity** | Code is divided into functions | Code is divided into classes and objects |
| **Security** | Less secure (no access control for data) | More secure (uses access specifiers like private) |
| **Reusability** | Limited code reusability | Promotes high code reusability through inheritance |
| **Example Language** | C, Pascal | C++, Java, Python (supports OOP) |

## 2. List and explain the main advantages of OOP over POP.

**Main Advantages of OOP over POP:**

1. **Encapsulation**: Keeps data safe from outside interference by wrapping it with methods inside classes.
2. **Reusability**: Inheritance allows reuse of existing code with enhancements or changes.
3. **Scalability**: Easier to manage large and complex programs due to modular structure.
4. **Maintainability**: Code is easier to update and maintain.
5. **Flexibility through Polymorphism**: Functions and operators can behave differently based on context.
6. **Real-World Modeling**: Objects can represent real-world entities, improving clarity and problem-solving.

## 3. Explain the steps involved in setting up a C++ development environment.

**Steps Involved in Setting Up a C++ Development Environment:**

**A. On Windows:**

1. **Install a Compiler:**
   * Install **MinGW** or **TDM-GCC** to get the g++ compiler.
2. **Install an IDE (optional but helpful):**
   * Examples: **Code::Blocks**, **Dev C++**, **Visual Studio**, or **VS Code**.
3. **Configure the IDE:**
   * Set the compiler path in IDE settings.
4. **Test Setup:**
   * Write a simple Hello World program.
   * Compile and run to verify setup.

## 4. What are the main input/output operations in C++? Provide examples.

**Main Input/Output Operations in C++ with Examples:**

**Header File Needed:**

**Example:**

#include <iostream>

using namespace std;

main()

{

int age;

string name;

cout << "Enter your name: ";

cin >> name;

cout << "Enter your age: ";

cn >> age;

cout << "Hello, " << name << "! You are " << age << " years old."

}

#include <iostream>

* **Input:** cin
* **Output:** cout