Theory Question

27\_jan\_se NAME: Vrinda Mavadhiya

## 1. What are the different data types available in C++? Explain with examples.

**Different Data Types in C++ (with Examples):**

C++ data types are mainly classified into **basic**, **derived**, **enumerated**, and **user-defined** types.

**A. Basic Data Types:**

| **Data Type** | **Description** | **Example** |
| --- | --- | --- |
| int | Integer numbers | int age = 25; |
| float | Single precision floating point | float price = 99.99; |
| double | Double precision floating point | double pi = 3.14159; |
| char | Single character | char grade = 'A'; |
| bool | Boolean (true or false) | bool isPassed = true; |
| void | No value (used for functions) | void greet() {} |

**B. Derived Data Types:**

* **Array**, **Pointer**, **Reference**, **Function**

cpp

CopyEdit

int nums[5]; // Array

int\* ptr = &age; // Pointer

**C. User-Defined Type:**

* **struct**, **class**, **union**

struct Student

{

int id;

string name;

};

## 2. Explain the difference between implicit and explicit type conversion in C++.

**Implicit vs Explicit Type Conversion:**

| **Type** | **Description** | **Example** |
| --- | --- | --- |
| **Implicit** | Done automatically by the compiler when compatible types are used | int x = 5; float y = x; |
| **Explicit** | Manually specified by the programmer using **type casting** | float x = 5.7; int y = (int)x; |

**Example:**

int x = 10;

float y = x;

float a = 9.81;

int b = (int a);

## 3. What are the different types of operators in C++? Provide examples of each.

**Types of Operators in C++ (with Examples):**

| **Operator Type** | **Example Code** |  |  | **Description** |
| --- | --- | --- | --- | --- |
| **Arithmetic** | + - \* / % |  |  | int c = a + b; |
| **Relational** | == != > < >= <= |  |  | if (a > b) |
| **Logical** | `&& |  |  |  |
| **Assignment** | = += -= \*= /= |  |  | x += 5; |
| **Increment/Decrement** | ++ -- |  |  | x++; --y; |
| **Bitwise** | `& |  |  | ^ ~ << >>` |
| **Ternary** | ? : |  |  | int max = (a > b) ? a : b; |
| **Scope Resolution** | :: |  |  | cout << ::x; (access global x) |
| **Sizeof** | sizeof() |  |  | cout << sizeof(int); |

## 4. Explain the purpose and use of constants and literals in C++.

**Purpose and Use of Constants and Literals in C++:**

**Constants:**

* Fixed values that **do not change** during execution.
* Declared using const or #define.

const float PI = 3.14;

#define MAX 100

**Literals:**

* Actual values assigned to variables or constants in the code.

**Types of Literals:**

* **Integer**: 10, -25
* **Floating-point**: 3.14, -0.99
* **Character**: 'A', 'z'
* **String**: "Hello", "C++"
* **Boolean**: true, false

**Usage Example:**

const int MAX\_USERS = 100;

string name = "Alice";

char grade = 'A';