🔸 1. **Variables in C++ (Java ke jaisa)**

| **Java Type** | **C++ Equivalent** |
| --- | --- |
| Local Variable | Function ke andar declare hota hai |
| Instance Variable | Class ke andar declare, non-static member |
| Static Variable | static keyword se class level variable |

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━

🔹 LOCAL vs GLOBAL VARIABLES IN C++

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━

✅ 1. Local Variable (Function ke andar banaye jaate hain)

📌 Definition:

Local variable wo variable hota hai jo kisi function, loop ya block ke andar banaya jaata hai.

Iska scope sirf usi block tak hota hai. Bahar use nahi kar sakte.

📦 Memory: Function chalte hi banta hai, aur function khatam hote hi destroy ho jaata hai.

🔹 Example:

#include <iostream>

using namespace std;

void printNumber() {

int x = 10; // local variable

cout << "Local x = " << x << endl;

}

int main() {

printNumber();

// cout << x; // ❌ Error: x not accessible here

return 0;

}

📝 Output:

Local x = 10

---

✅ 2. Global Variable (Function ke bahar declare kiye jaate hain)

📌 Definition:

Global variable wo hota hai jo sabhi functions ke bahar declare kiya jaata hai.

Ye pura program me kahin bhi accessible hota hai (agar scope me ho).

📦 Memory: Program start hote hi banta hai, aur program end hone tak memory me rehta hai.

🔹 Example:

#include <iostream>

using namespace std;

int x = 100; // global variable

void show() {

cout << "Global x in show(): " << x << endl;

}

int main() {

cout << "Global x in main(): " << x << endl;

show();

return 0;

}

📝 Output:

Global x in main(): 100

Global x in show(): 100

---

🔁 Difference Table:

| Feature | Local Variable | Global Variable |

|--------------------|------------------------|-----------------------------|

| Kahan banta hai? | Function/block ke andar | Function ke bahar |

| Scope | Sirf block ke andar | Pura program me access hota hai |

| Memory lifetime | Function ke time tak | Program end tak |

| Access outside block | ❌ Nahi ho sakta | ✅ Ho sakta hai |

---

💡 Note:

- Agar local aur global variable dono ka naam same ho to local variable ko priority milti hai.

- Aap `::` operator se global variable ko forcefully access kar sakte ho.

🔹 Example:

#include <iostream>

using namespace std;

int x = 50; // global

int main() {

int x = 10; // local

cout << "Local x = " << x << endl; // Output: 10

cout << "Global x = " << ::x << endl; // Output: 50

return 0;

}

🡺Static Varible : 🡺   
  
#include <iostream>

using namespace std;

void counter() {

static int count = 0; // static local variable

count++;

cout << "Count: " << count << endl;

}

int main() {

counter(); // Output: Count: 1

counter(); // Output: Count: 2

counter(); // Output: Count: 3

return 0;

}



🡺