پاسخ سوال دو

تفاوت اصلی بین تابع با آرگومان و ارجاع و تابع با آرگومان اشاره گر در زبان C++ در روش انتقال آرگومان‌ها به تابع است.

تابع با آرگومان: در این حالت، مقادیر آرگومان‌ها به تابع انتقال داده می‌شوند. به عبارت دیگر، مقادیر کپی از آرگومان‌ها به پارامترهای تابع منتقل می‌شوند. هر گونه تغییر در مقدار پارامترهای تابع، تنها بر روی کپی از آرگومان‌ها تاثیر می‌گذارد و اصلی‌ها تغییرات گسترده نمی‌کنند.

مثال:

void incrementByRef(int& x) {

x++;

}

int main() {

int num = 5;

incrementByRef(num);

std::cout << num; // خروجی: 6

return 0;

}

در این مثال، مقدار num به تابع increment انتقال داده می‌شود اما تابع فقط روی کپی آن عملیات انجام می‌دهد. بنابراین، تغییر در شماره در تابع increment تأثیری در مقدار اصلی num در تابع main ندارد.

تابع با آرگومان ارجاع: در این حالت، آرگومان‌ها به عنوان ارجاع یا شبیه‌سازی از آرگومان‌ها به تابع انتقال داده می‌شوند. به عبارت دیگر، نشانگر یا اشاره‌گر به آرگومان‌ها در تابع کپی می‌شود و تغییر در آن‌ها تأثیری در مقدار اصلی آرگومان دارد.

مثال: void incrementByRef(int& x) {

x++;

}

int main() {

int num = 5;

incrementByRef(num);

std::cout << num; // خروجی: 6

return 0;

}

در این مثال، آرگومان num به تابع incrementByRef انتقال داده می‌شود و تابع به‌جای کپی آرگومان، نشانگر آن را دریافت می‌کند. بنابراین، هر تغییر در آرگومان x که یک ارجاع به num است، تأثیری در مقدار اصلی num در تابع main دارد.

تابع با آرگومان اشاره‌گر: در این حالت، آرگومان‌ها به عنوان اشاره‌گر به آرگومان‌ها در تابع انتقال داده می‌شوند. به عبارت دیگر، اشاره‌گر به آرگومان‌ها در تابع کپی می‌شود و تغییر در آن‌ها تأثیری در مقدار اصلی آرگومان دارد.

مثال:

void incrementByPtr(int\* x) {

(\*x)++;

}

int main() {

int num = 5;

incrementByPtr(&num);

std::cout << num

// خروجی: 6

return 0;

}

در این مثال، آرگومان num به تابع incrementByPtr با یک اشاره‌گر می‌شود. در تابع incrementByPtr، آرگومان به عنوان یک اشاره‌گر کپی می‌شود و با استفاده از این اشاره‌گر، می‌توان به مقدار اصلی num دسترسی پیدا کرد و تغییری در آن داد. بنابراین، در این مثال نیز تغییر در آرگومان x (که یک اشاره‌گر به num است) تأثیری در مقدار اصلی num دارد.