Cấp phát tĩnh

Định nghĩa: Biến được cấp phát cố định trong quá trình biên dịch và tồn tại trong vùng nhớ tĩnh (static memory).

Ví dụ: Các biến static hoặc biến toàn cục.

Thời điểm cấp phát: Tại compile-time, bộ nhớ được phân bổ và giữ nguyên trong suốt vòng đời của chương trình.

Ưu điểm:

Nhanh hơn vì vùng nhớ được cố định và không cần phải tìm chỗ trống trong bộ nhớ khi chạy.

Nhược điểm:

Không linh hoạt vì kích thước được cố định tại compile-time.

Cấp phát động

Định nghĩa: Bộ nhớ được cấp phát tại thời điểm chạy (run-time) bằng cách tạo các đối tượng hoặc mảng.

Thời điểm cấp phát: Khi new được sử dụng để tạo một đối tượng/mảng, bộ nhớ sẽ được cấp phát trong heap.

Ưu điểm:

Linh hoạt hơn vì kích thước có thể được xác định trong lúc chạy.

Nhược điểm:

Chậm hơn cấp phát tĩnh vì cần tìm và quản lý bộ nhớ trong heap.

Bộ nhớ Stack

Chức năng:

Lưu trữ các biến cục bộ, tham số của phương thức, và con trỏ tới đối tượng trên heap.

Quản lý:

Tuân theo nguyên tắc LIFO (Last In, First Out) – biến sẽ được giải phóng ngay khi phương thức kết thúc.

Tốc độ:

Nhanh vì bộ nhớ được giải phóng tự động sau mỗi lần thoát khỏi phương thức.

Nhược điểm:

Kích thước của stack có giới hạn.

Chỉ lưu trữ dữ liệu có vòng đời ngắn.

Bộ nhớ Heap

Chức năng:

Dùng để lưu trữ đối tượng và biến tham chiếu được tạo bằng từ khóa new.

Quản lý:

Bộ nhớ không tự động giải phóng khi đối tượng không dùng nữa mà phải thông qua Garbage Collector.

Tốc độ:

Chậm hơn stack vì cần thời gian để tìm và giải phóng bộ nhớ.

Ưu điểm:

Lưu trữ dữ liệu có vòng đời dài hơn (tồn tại cho đến khi bị thu gom rác).

Nhược điểm:

Cần thời gian để quản lý và dọn dẹp thông qua Garbage Collector.