# Cấp phát tĩnh và cấp phát động

1. **Cấp phát tĩnh (Static Allocation)**

* Cấp phát tĩnh xảy ra khi biến được khai báo ở cấp độ class hoặc biến toàn cục (global variable). Bộ nhớ cho các biến này được cấp phát khi chương trình khởi động và sẽ tồn tại cho đến khi chương trình kết thúc.
* Ví dụ: Biến static, biến toàn cục.

1. **Cấp phát động (Dynamic Allocation)**

* Cấp phát động xảy ra khi bộ nhớ được cấp phát trong quá trình thực thi chương trình, thường sử dụng các phương thức như ‘new’ để tạo object.
* Ví dụ: object được tạo bằng từ khóa ‘new’.

# Bộ nhớ heap và bộ nhớ stack

1. **Bộ nhớ heap**

* Bộ nhớ heap được sử dụng để lưu trữ các object và dữ liệu có thời gian sống dài hơn, không bị giới hạn bởi phạm vi hàm. Bộ nhớ heap được quản lý thông qua Garbage Collection trong Java.
* Cần phải giải phóng bộ nhớ khi không còn cần thiết (được tự động xử lý bởi Garbage Collector).
* Chậm hơn so với stack do cần phải tìm vị trí trống để cấp phát bộ nhớ.
* Có thể cấp phát bộ nhớ lớn hơn.
* Ví dụ: Đối tượng được tạo bằng từ khóa ‘new’.

1. **Bộ nhớ stack**

* Bộ nhớ stack được sử dụng để lưu trữ các biến cục bộ và các thông tin liên quan đến gọi hàm (function call). Mỗi khi một hàm được gọi, một khung ngăn xếp (stack frame) mới được tạo ra và khi hàm đó kết thúc, khung ngăn xếp sẽ bị xóa.
* Không cần phải giải phóng bộ nhớ.
* Nhanh chóng trong việc truy xuất và cấp phát.
* Nếu kích thước ngăn xếp vượt quá, sẽ xảy ra lỗi tràn ngăn xếp (stack overflow).
* Ví dụ: Biến cục bộ trong một hàm.