

- حافظه Stack:

حافظه Stack یک نوع حافظه در کامپیوتر است که برای ذخیره داده‌ها و آدرس‌های بازگشتی از توابع استفاده می‌شود. در این حافظه، داده‌ها به صورت پشته‌ای (Last In, First Out) ذخیره می‌شوند، به این معنا که آخرین داده‌ای که وارد شده، اولین داده‌ای است که خارج می‌شود. هر زمان یک تابع صدا زده می‌شود، یک فریم جدید در Stack ایجاد می‌شود که شامل متغیرهای محلی و آدرس بازگشتی آن تابع است. زمانی که تابع از اجرا خارج می‌شود، فریم مربوطه از Stack حذف می‌شود و کنترل به آدرس بازگشتی باز می‌گردد.

- حافظه Heap:

حافظه Heap یک منطقه دیگر از حافظه کامپیوتر است که برای ذخیره داده‌هایی که به صورت پویا ایجاد و مدیریت می‌شوند، استفاده می‌شود. در این حافظه، داده‌ها به صورت دسترسی تصادفی (Random Access) ذخیره می‌شوند، یعنی می‌توان به هر زمان به هر مکان در حافظه دسترسی داشت. حافظه Heap از دیدگاه برنامه نویسی، مکانی است که شیء‌ها و داده‌هایی که با استفاده از new یا malloc ایجاد می‌شوند، در آنجا ذخیره می‌شوند. این حافظه باید به صورت صریح توسط برنامه نویس مدیریت شود و باید دقت شود که داده‌هایی که در آنجا ذخیره می‌شوند مناسباً رها نشوند تا از memory leak جلوگیری شود.

- تفاوت:

1. استفاده: حافظه Stack برای ذخیره داده‌ها و آدرس‌های بازگشتی از توابع استفاده می‌شود، در حالی که حافظه Heap برای ذخیره داده‌های پویا ایجاد شده توسط برنامه‌نویس (مانند شیء‌ها) استفاده می‌شود.
2. مدیریت: حافظه Stack به صورت خودکار توسط کامپایلر مدیریت می‌شود و داده‌ها به صورت LIFO ذخیره و حذف می‌شوند، در حالی که حافظه Heap نیاز به مدیریت دستی دارد و باید برنامه‌نویس از آن دقت کند که داده‌ها را صحیحاً آزاد کند تا از memory leak جلوگیری شود.
3. سرعت دسترسی: دسترسی به حافظه Stack سریع‌تر است زیرا داده‌ها به صورت مرتب در آنجا ذخیره می‌شوند و دسترسی به آنها ساده‌تر است، در حالی که دسترسی به حافظه Heap کمی زمان‌برتر است زیرا داده‌ها به صورت پراکنده در آنجا ذخیره می‌شوند و نیاز به پیدا کردن مکان صحیح دارد.