Last Week

- Pointer

Summary

- **&**A: gives the (memory) **address** of (a variable) A.
- **Pointer** variable: let *int *ptr_A* = **&**A
  - *ptr_A* contains the **address**, on the other hand,
  - **ptr_A* gives the **actual value** of the data to which the pointer is pointing, i.e. *de-referencing*.

```
int A;
```

```
int *ptr;
ptr = &A;
```

==

```
int *ptr = &A;
```

Keyboard input: scanf, scanf_s()

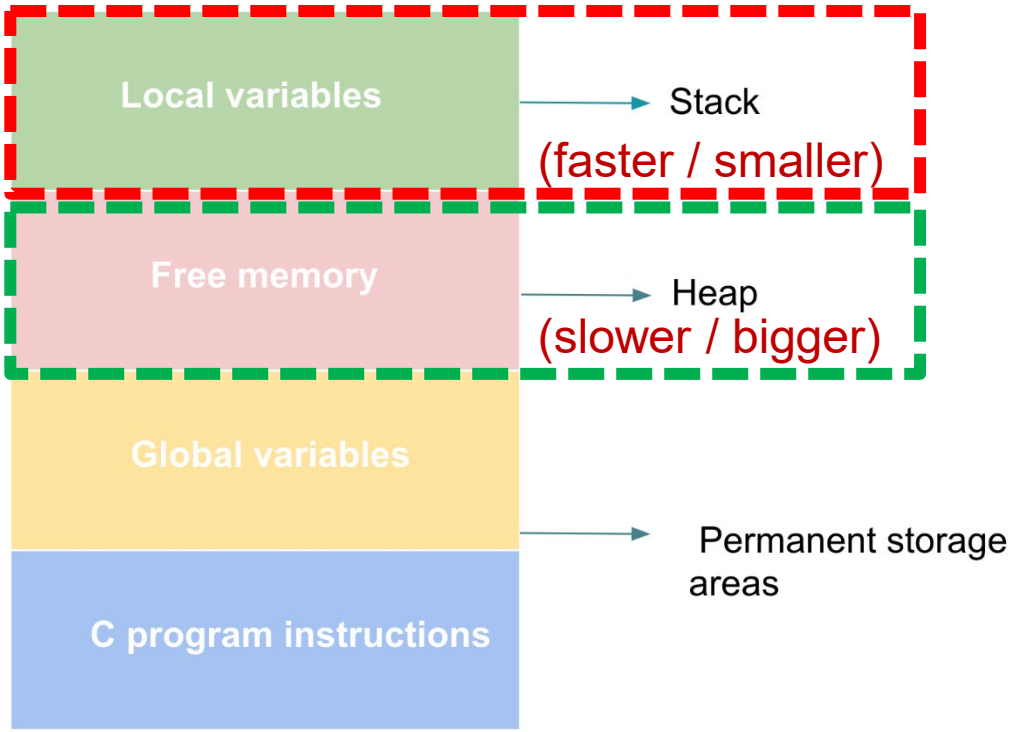
scanf_s("*input data type*", *address of the variable*) // new compilers
scanf() // for old C compilers, e.g., NewbieIDE, Code::Block, Xcode

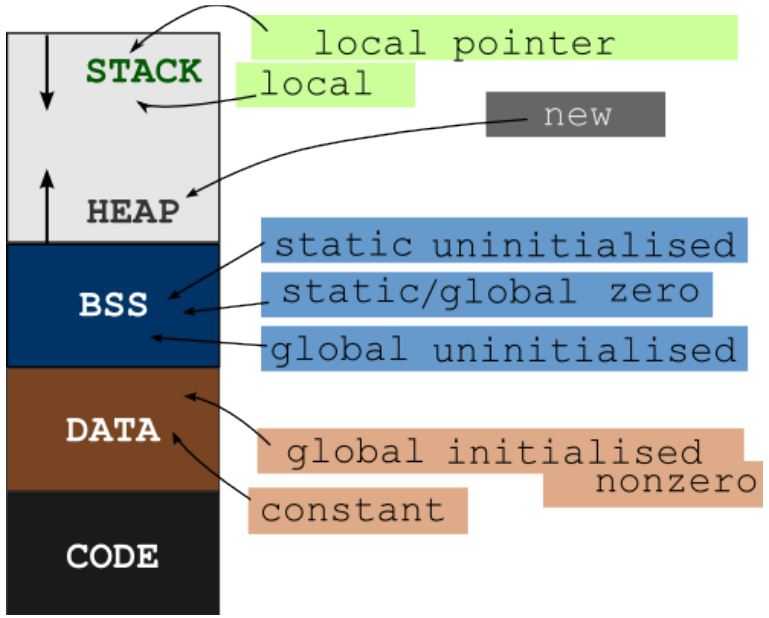
```
int page;
printf("Page number: ");
scanf_s("%d", &page);   // Visual Studio
                        // or scanf  for old C compilers

printf("Your page number: %d \n", page);
```

Memory Segments in C

- A ***stack*** is used for ***static*** memory allocation and a ***heap*** for ***dynamic*** memory allocation, both stored in your computer's **RAM**.
- ***Dynamic*** memory allocation: refer to "Dynamic Memory Allocation" on Resources, LEARN.





Last Week

String (a char **array**)


```
char author0[] = "Li"; // as we did it before

// (1) name of an array is a pointer variable
// (2) name of an array == address of the element 0

printf("\n %c \t %c", *author0, *(author0 + 1));
// *(name + n) will return the data stored in the n position
// Or,
printf("\n %c \t %c", author0[0], author0[1]);
```