SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

19ECB211 – Microcontroller Programming & Interfacing

II YEAR/ IV SEMESTER

UNIT 3 – PIC PROGRAMMING IN C

TOPIC 7 – Data RAM allocation in C

Data RAM allocation in C

- Data RAM allocation in C refers to the process of allocating memory space in the Random Access Memory (RAM) of a computer system to store data during program execution

- In C, there are primarily two types of RAM allocation: static allocation and dynamic allocation

Data RAM allocation in C

- Data RAM allocation in C refers to the process of allocating memory space in the Random Access Memory (RAM) of a computer system to store data during program execution

- In C, there are primarily two types of RAM allocation: static allocation and dynamic allocation

Data RAM allocation in C

Static Allocation

- Static allocation is done at compile time

- Memory for variables is allocated when the program starts and persists throughout its execution

- Variables declared outside of functions (global variables) or with the **static** keyword within functions are statically allocated

eg.

```
 int globalVariable; // Statically allocated global variable
 static int staticVariable; // Statically allocated static variable
```

Data RAM allocation in C

Dynamic Allocation:

- Dynamic allocation allows memory to be allocated during runtime

- Memory is allocated from the heap segment of RAM using functions like **malloc()**, **calloc()**, or **realloc()**

- Dynamic allocation provides flexibility in memory usage but requires explicit memory management by the programmer

Eg

```
int *dynamicVariable;
dynamicVariable = (int *)malloc(sizeof(int)); // Dynamic allocation
```

Data RAM allocation in C

Stack Allocation

- Stack allocation is used for local variables within functions
- Memory for stack-allocated variables is automatically managed by the compiler
- Variables are allocated and deallocated in a last-in, first-out (LIFO) manner
- Stack allocation is fast but limited in size and scope

Eg

```
 void someFunction() {
 int stackVariable; // Stack-allocated variable
// ...
 }
```

Data RAM allocation in C

Heap Allocation vs. Stack Allocation

- Heap allocation is suitable for large or dynamically-sized data structures, whereas stack allocation is preferable for smaller, short-lived variables

- Heap allocation requires manual memory management (allocation and deallocation) by the programmer, while stack allocation is managed automatically by the compiler

- Improper use of heap allocation can lead to memory leaks or fragmentation, while stack allocation is generally safer and more efficient for managing local variables

- Understanding data RAM allocation in C is crucial for efficient memory management and optimizing the performance of C programs. Programmers must choose the appropriate allocation method based on the specific requirements of their applications

THANK YOU