Programming Paradigms

CT331 Lecture 4

Finlay Smith

finlay.smith@universityofgalway.ie

User Defined Types: typedef

Defining a type with typedef

```
typedef existing_name alias_name;
```

EG:

```
typedef unsigned char BYTE;   // creates new type.

BYTE b; //creates a variable of type BYTE

b = 10; //stores the number 10 in BYTE b.
```

Defining a type with typedef

```
struct studentStruct {
       char name[30];  //student name
       int number;  //student id number
};

typedef struct studentStruct student;

// we can now refer to struct studentStruct as student
```

Defining a type with typedef

```
// we can combine typedef and struct...

typedef struct studentStruct {
       char name[30];  //student name
       int number;  //student id number
} student;

// we can now refer to struct studentStruct as student
```

More Memory Allocation

Memory Allocation Functions

There are 4 functions that can be used for dynamic memory allocation:

- malloc()
- calloc()
- free()
- realloc()

Memory Allocation Functions

malloc() is the simplest memory allocation function:

- Takes a single argument
    - The number of bytes to allocate
- Returns a pointer to the allocated memory

Memory Allocation Functions

calloc() is a slightly more advanced memory allocation function:

- Takes a two arguments
  - The number of elements to be stored in allocated memory
  - The size of each of those element
- Returns a pointer to the allocated memory

Memory Allocation Functions

calloc():

- Also assigns the value 0 to each of the elements
  - Unlike malloc()
- Useful for dynamically creating arrays

Memory Allocation Functions

free() releases memory, allowing it to be reused :

- Takes a single argument
    - A pointer to the memory being released
- No return value
- Does not delete values stored in that memory, nor remove access to the memory

Memory Allocation Functions

realloc() changes size of a referenced block of memory:

- Takes a two arguments
    - A pointer to the memory currently referenced
    - The number of bytes now to be pointed to
- Returns pointer to new block of memory