Header Files, Guards and Macros

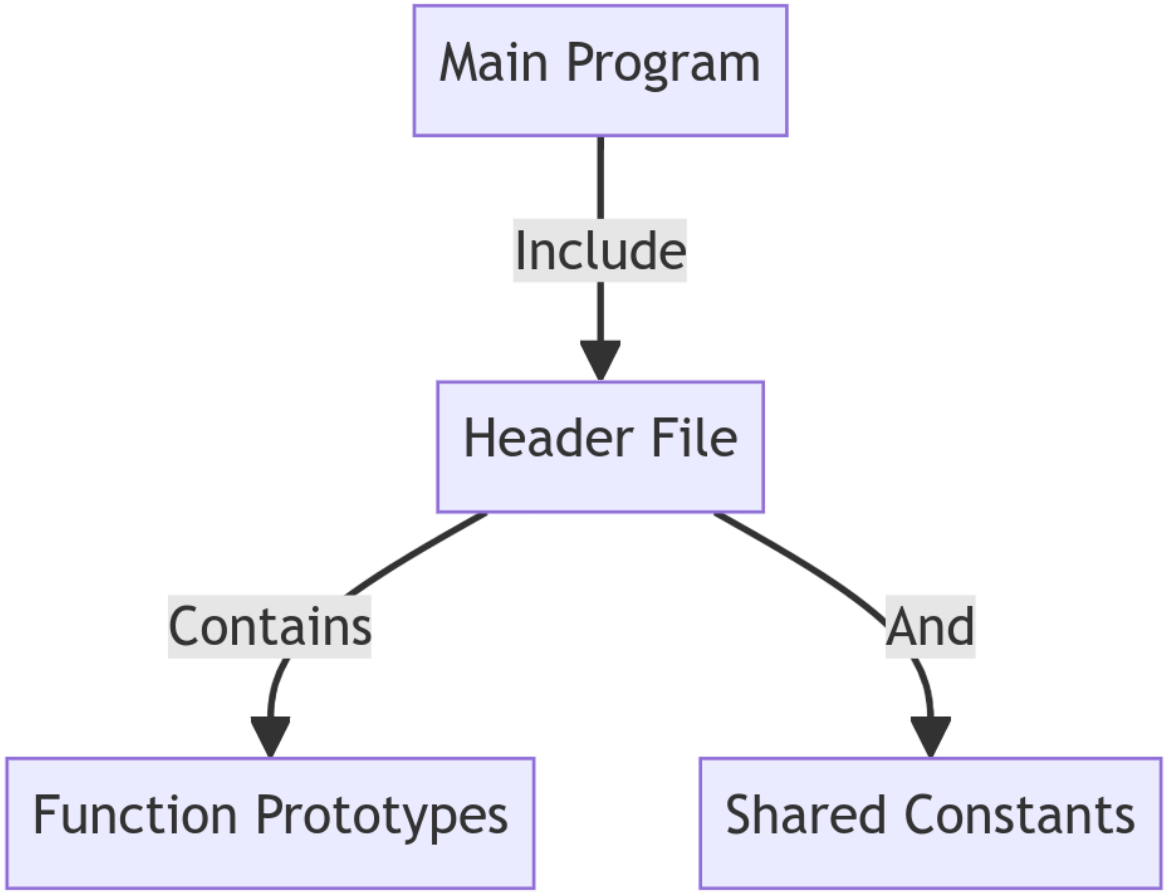
```
Module Code: ELEE1147

Module Name: Programming for Engineers

Credits: 15

Module Leader: Seb Blair BEng(H) PGCAP MIET MIHEEM FHEA
```

Download as a PDF

Why Use Header Files?

- **Modularity:** Separate interface from implementation.
- **Reusability:** Share functions and data structures.
- **Readability:** Enhance code organisation.
- **Function Prototypes:** Allows the compiler to check function signatures during compilation.
- **Precompiled Headers:** Speeds up compilation by avoiding redundant parsing of headers in multiple source files.

How Does It Work?



Header Example

```c
// header.h
#ifndef HEADER_H // Header guard
#define HEADER_H // Macro

#include <stdio.h> // Other libraries

void greeter(); // Function prototype

#define PI 3.14159 // Shared constant
#define GR ((double)1.61803) // ""

// Shared DataStorage
struct Student {
  char name[50];
  int studentId;
  float classification;
};
#endif // HEADER_H
```

```c
// main.c
#include "header.h"
int main() {
    greeter();
    return 0;
}
```

```c
#include "header.h"

void greeter(){
    printf("Hello World!")!
}
```

What are Header Guards?

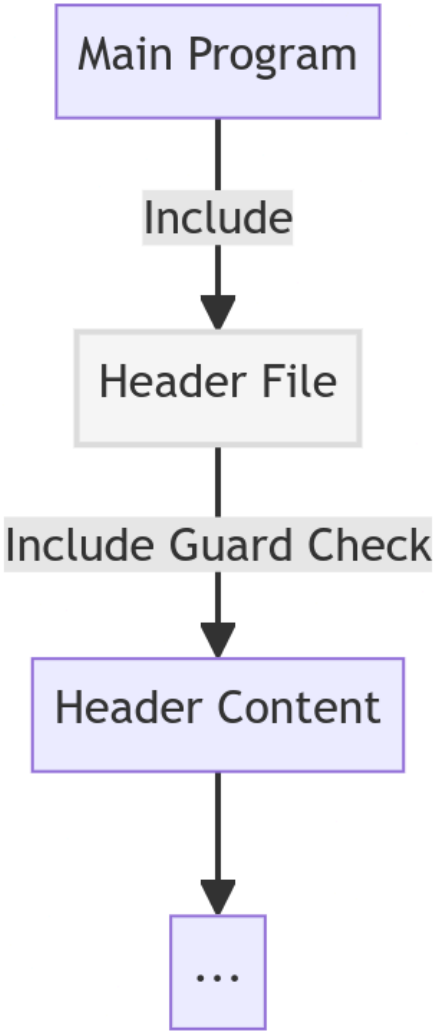
- **Purpose:** Prevent multiple inclusions of the same header file.
- **Issue:** Without guards, redefinitions can occur during multiple inclusions.
- **Solution:** Use preprocessor directives to conditionally include the contents.

Why Use Header Guards?

- **Avoid Redefinitions:** Prevent compilation errors due to duplicate declarations.
- **Ensure Once-Only Inclusion:** Each header is included only once in a translation unit.
- **Improve Compilation Efficiency:** Reduce redundant parsing of header contents.

```
#ifndef HEADER_H
#define HEADER_H

...

#endif //end of HEADER_H
```

How Header Guards Work



Macros

Macros in C are a way to **define** constants or simple functions using the `#define` directive. They are preprocessor directives, meaning they are processed before the actual compilation of the code.

```
// example_macros.h

#ifndef EXAMPLE_MACROS_H
#define EXAMPLE_MACROS_H

#define PI 3.14159 // Shared Constant
#define SQUARE(x) ((x) * (x)) // Function

#ifdef _MSC_VER
// Code specific to Microsoft Version C/C++
#endif //end of _MSC_VER

#endif // end of EXAMPLE_MACROS_H
```

Preprocessor Directive: `#include ""` vs `<>`

- Use `#include ""`
  - for including header files that are part of your project or are in the current directory.
- Use `#include <>` for
  - including standard library header files or other headers that are part of the system include directories.

Standardised Header Examples:

stdio.h

```
23. #ifndef _STDIO_H
24. #define _STDIO_H    1
25.
26. #define __GLIBC_INTERNAL_STARTING_HEADER_IMPLEMENTATION
27. #include <bits/libc-header-start.h>
...
878.
879. __END_DECLS
880.
881. #endif /* <stdio.h> included.  */
```

Standardised Header Examples:

math.h

```
...
130. #define    M_E             ((double)2.71828182845904523536)  /* e */
131. #define    M_LOG2E         ((double)1.44269504088896340736)  /* log 2e */
132. #define    M_LOG10E        ((double)0.43429448190325182765) /* log 10e */
133. #define    M_LN2           ((double)0.69314718055994530942) /* log e2 */
134. #define    M_LN10          ((double)2.30258509299404568402) /* log e10 */
135. #define    M_PI            ((double)3.14159265358979323846) /* pi */
...
494. int __signbitl(long double);
495. __END_DECLS
496.
497. #endif /* !_MATH_H_ */
```


INCLUDING A HEADER FILE IN C BE LIKE..
TA-DA! JUST SOME DECLARATIONS OUT OF THIN AIR!