

EE 200 Lecture 3: Writing code

Steven Bell

13 September 2022



Macros

Are text substitutions done by the preprocessor

```
#define BIG_NUM 9000;
if(score > BIG_NUM){
    printf("great job!\n");
}
```

```
#define BIG_NUM 9000;
if(score > 9000;){
    printf("great job!\n");
}
```

Macros

```
// Higher levels print more things; 0 turns everything off
#ifndef DEBUG_LEVEL
#define DEBUG_LEVEL 5
#endif
#define FATAL(...) dbgprint(0, __FILE__, __LINE__, __VA_ARGS__)
#define ERROR(...) dbgprint(1, __FILE__, __LINE__, __VA_ARGS__)
#define WARN(...)  dbgprint(2, __FILE__, __LINE__, __VA_ARGS__)
#define DEBUG(...) dbgprint(3, __FILE__, __LINE__, __VA_ARGS__)
#define LOG(...)   dbgprint(4, __FILE__, __LINE__, __VA_ARGS__)
#define TRACE(...) dbgprint(5, __FILE__, __LINE__, __VA_ARGS__)
void dbgprint(int level, const char* file, int line, const char* fmt, ...);
```

#ifdef

A preprocessor if-statement

```
#ifdef LINUX
```

```
    // Magic stuff that only works on Linux
```

```
#endif
```

```
#ifdef AARCH64
```

```
    // Magic stuff that only works on 64-bit ARM processors
```

```
#endif
```

```
#ifdef 0
```

```
    // Code that I don't want to compile
```

```
#endif
```

Protection macro

problem1.h

```
#ifndef PROBLEM1_H
#define PROBLEM1_H
// Contents of the header file
#endif
```

problem1_helpers.h

```
#ifndef PROBLEM1_HELPERS_H
#define PROBLEM1_HELPERS_H
#include "problem1.h"
// Contents of the header file
#endif
```

test_problem1.c

```
#include "problem1.h"
#include "problem1_helpers.h"
int main(int argc, char* argv[])
{
    // Stuff...
}
```

Using git

git clone

git add

git commit

gitk

Compiling code

```
gcc problem1.c test_problem1.c -o prob1
```

```
int isMultipleOfTen(int value)
{
    int div10 = value / 10;
    if(value == 10*div10){
        return 1;
    } else {
        return 1;
    }
}
```

```
int main(int argc, char* argv[])
{
    assert(isMultipleOfTen(70) == 1);
    assert(isMultipleOfTen(10) == 1);
    printf("All tests passed.\n");
    return(0);
}
```



```
int isMultipleOfTen(int value)
{
    int div10 = value / 10;
    if(value == 10*div10){
        return 1;
    } else {
        return 1;
    }
}
```

(Highlighted code is executed)

```
int main(int argc, char* argv[])
{
    assert(isMultipleOfTen(70) == 1);
    assert(isMultipleOfTen(10) == 1);
    printf("All tests passed.\n");
    return(0);
}
```

```
int isMultipleOfTen(int value)
{
    int div10 = value / 10;
    if(value == 10*div10){
        return 1;
    } else {
        return 1;
    }
}
```

```
int main(int argc, char* argv[])
{
    assert(isMultipleOfTen(70) == 1);
    assert(isMultipleOfTen(10) == 1);
    assert(isMultipleOfTen(3) == 0);
    printf("All tests passed.\n");
    return(0);
}
```

Collaboration policy

Skipping to chapter 8 in AoP

Classwork 3 is hosted on Github

You have your very own repository:

```
git clone https://github.com/tuftsee200
```