CME 331 – Embedded Systems – Fall 2025
Assignment 1

This assignment contains 4 sections of C code topics. The intent is that you determine your answers via code inspection. Completed assignments must be submitted as a .pdf, on Canvas, by the specified due date. Late or missed assignments will not be marked (i.e., graded as zero).

---

1. Operators – consider the following code:
(hint: you should know what an electrical short-circuit is, but do you know what a logic short-circuit is?)

```
#include <stdio.h>

int main()
{
  int a = 1;
  int b = 2;
  int c = !a || b;
  int d = a || --b;
  int e = a-- && --b;
  printf("a = %d, b = %d, c = %d, d = %d, e = %d", a, b, c, d, e);
  return 0;
}
```

What is the output of the program? (5 points)

a = , b = , c = , d = , e =

2. Preprocessor directives – consider the following code:

```
#include <stdio.h>
#define square(num) num*num

int main()
{
  int val1, val2;
  val1 = 16 / square(4);
  val2 = 16 - square(4);
  printf("Value 1 = %d, Value 2 = %d", val1, val2);
  return 0;
}
```

a) What is the output of this program? (2 points)

Value 1 = , Value 2 =

b) How does preprocessing alter the lines calculating val1 and val2? (2 points)

val1 =

val2 =

c) How could the define statement be improved to result in more reliable behaviour? (1 point)

#define square(num) _____

3. Bitmasking – determine the output of the printf statements

a) printf("Ans: %d", 13 | (1 << 5)); (1 point)

Ans:

b) printf("Ans: %X", (0x4000D00D | 0xA)); (1 point)

Ans:

c) printf("Ans: %d", 80 & ~16); (1 point)

Ans:

d) Consider the following code and determine the output of the printf statement (2 points)

```
#include <stdio.h>
#define ASSIGN_R 0x4A4253FC

int main()
{
  long val1, val2;
  val1 = ASSIGN_R;
  val1 &= ~0x77;
  val2 = val1;
  val2 ^= 0xFF;
  printf("Value 1: %X, Value 2: %X", val1, val2);
  return 0;
}
```

Value 1: Value 2:

4. Pointers

a) Consider the following code and determine the output of the printf statement (1 point)

```
#include <stdio.h>

void func(int *ptr)
{
  *ptr = 100;
}

int main()
{
  int val = 5;
  func(&val);
  printf("Ans: %d", val);
  return 0;
}
```

Ans:

b) Consider the following code and determine the output of the printf statement (1 point)

```
#include <stdio.h>

void func(int *ptr, int num)
{
  num = num + 15;
  *ptr = *ptr + num;
  return;
}

int main()
{
  int val1 = 10, val2 = 25;
  func(&val1,val2);
  printf("Ans: %d", val1+val2);
  return 0;
}
```

Ans:

c) Consider the following code and determine the output of the printf statement (2 points)

```
#include <stdio.h>

void func(int *ptr1, int *ptr2)
```

```
{
  ptr1 = ptr2;
  *ptr1 = 2;
  return;
}

int main()
{
  int val1 = 10, val2 = 20;
  func(&val1,&val2);
  printf("Value 1: %d, Value 2: %d", val1, val2);
  return 0;
}
```

Value 1: Value 2:

d) Consider the following code and determine the output of the printf statement (1 point)

```
#include <stdio.h>

int main()
{
  int *ptr;
  int val;

  ptr = &val;
  *ptr = 0;
  *ptr += 5;
  printf("Value 1: %d", val);
  return 0;
}
```

Value 1: