ENEE 140, Spring 2016
Midterm Exam — Answer Key
Do Not Make a Copy!!

Date: Tuesday, March 29, 2016, 3:30 – 4:45 pm

University of Maryland Honor Pledge: The University is committed to Academic Integrity, and has a nationally recognized Honor Code, administered by the Student Honor Council. In an effort to affirm a community of trust, the Student Honor Council proposed and the University Senate approved an Honor Pledge. The University of Maryland Honor Pledge Reads:

> "*I pledge on my honor that I have not given or received any unauthorized assistance on this examination (or assignment)*"

Please write the exact wording of the Pledge, followed by your signature, in the space below:

Pledge: ______________________________
Pledge: ______________________________
Pledge: ______________________________
Pledge: ______________________________

Your signature: ______________________________
Full name: ____________________ Course: ________ Directory ID: ______________

List of Exam Questions:

Question:	1	2	3	4	5	6	Total
Points:	28	10	10	20	12	20	100
Score:							

Instructions:

- Make sure that your exam is not missing any sheets, then write your full name, your section and your Directory ID on the front.
- Write your answers in the space provided below the problem. If you make a mess, clearly indicate your final answer.

- The exam has a maximum score of 100 points.
- The problems are of varying difficulty. The point value of each problem is indicated. Pile up the easy points quickly and then come back to the harder problems.
- This exam is OPEN BOOK. You may use any books or notes you like. Calculators are allowed, but no other electronic devices. Good luck!

1. (28 points) This problem tests your understanding of C types and casts and of C operators. Assume that variables a, b, c and d are defined as follows:

```
int          a = 1;
unsigned     b = 2;
float        c = 3;
float        d = 4;
int          e = 5;
int          f = 2;
int          g = 0;
float        i = 8;
```

Fill in all the empty cells in the table below. For each of the C assignment expressions in the left column, state the resulting value of the r2–r9 variables. If an expression results in an error, write ERROR.

Assignment		**Value**
float	r1 = c/d;	0.75
int	r2 = a/b;	**0**
int	r3 = a%(**int**)d;	**1**
float	r4 = b/4;	**0**
int	r5 = (c/2 + a) * 3;	**7**
unsigned	r6 = ((**unsigned**)INT_MAX + a) % 2;	**0**
unsigned	r7 = (**unsigned**)INT_MIN * 2;	**0**
int	r8 = −INT_MIN > 0;	**0**
unsigned	r9 = (**unsigned**)d % b;	**0**
double	r10 = f / e * i + e;	**5**
double	r11 = e + i * f / e;	**8.2**
int	r12 = e * i % f;	**ERROR**
int	r13 = e || g && f;	**1**
double	r14 = e * i + − + f;	**38**
double	r15 = e + i * + * f;	**ERROR**

2. (10 points) This problem tests your understanding of C syntax. Find and correct each of the bugs in the program. Determine which bugs will result in compilation errors (the program will not compile) and which are logical errors that do not prevent the program from compiling.

```
#include <stdioh>
int main(void);
{
  int i;
  char c;
  float f = 2.0
  printf(”%i\n”, f);
  scanf(”%d␣%c␣%f”, i, c);
}
```

Solution:
#include <stdioh> should be stdio.h (error)
int main(**void**); remove ; (error)
float f = 2.0 add ; (error)
printf(”%i\n”, f); %i should be %f
scanf(”%d␣%c␣%f”, i, c); too many format specifiers, need &i, &c

3. (10 points) This problem tests your understanding of `for` and `while loops`. Rewrite the following program so that the only type of loop it uses are `while` loops.

```
#include <stdio.h>

int main()
{
  int i = 0, j = 0, k = 0;
  for (i = 0; i < 4; i++){
    for(j = 0; j < 10; j++) {
        printf(”#”);
    }
    printf(”\n”);
  }
  for (k = 10; k > -5; k--)
    printf(”The␣number␣is␣%d.\n”,k);
  return 0;
}
```

Solution:

```
#include <stdio.h>
```

```
int main()
{
  int i = 0, j = 0, k = 0;
  while (i < 4){
    j = 0;
    while (j < 10) {
        printf("#");
        j++;
    }
    printf("\n");
    i++;
  }
  k = 10;
  while(k > -5) {
    printf("The␣number␣is␣%d.\n",k);
    k--;
  }
  return 0;
}
```

4. (20 points) This problem tests your understanding of control flow. Fill in the blanks below to implement a function that computes the minimum of 3 numbers:

```
int
min3(int a, int b, int c)
{
    if (a ___< or <=___ b) {
        if (c < a) {
            return c;
        } else {
            return ____a____;
        }
    ____}____ else {
        if (c < ____b____) {
            return c;
        } else {
            return ____b____;
        }
    }
}
```

5. (12 points) This problem tests your understanding of arrays. What does this function do?

```
int
f1(int arr[], int arr_size)
```

```
{
        int x = arr[0], i;

        for (i = 1; i < arr_size; i++) {
                if (arr[i] < x) {
                        x = arr[i];
                }
        }
        return x;
}
```

Solution: Finds the smallest element in the array

6. (20 points) This problem tests your understanding of input/output and of control flow. Determine the output of the following program:

```
#include <stdio.h>

int multiply(int i);

int main(void)
{
  int i = 1;
  printf("multiply(i)␣=␣%d\n", multiply(i)%9);
  printf("%d\n", i);
}

int multiply(int i)
{
        int count = 0;
        printf("%d,␣%d", i*=100, ++count);
        return i;
}
```

Solution:

```
100,␣1multiply(i)␣=␣1
1
```