16.216: ECE Application Programming
Fall 2012

Exam 1
October 3, 2012

Name: _____ **ID #:** _____

For this exam, you may use only one 8.5" x 11" double-sided page of notes. All electronic devices (e.g., calculators, cellular phones, PDAs) are prohibited. If you have a cellular phone, please turn it off prior to the start of the exam to avoid distracting other students.

The exam contains 3 questions for a total of 100 points. Please answer the questions in the spaces provided. If you need additional space, use the back of the page on which the question is written and clearly indicate that you have done so.

Please read each question carefully before you answer. In particular, note that:
- Question 3 has three parts, but you are only required to complete two of the three parts.
  - You may complete all three parts for up to 10 points of extra credit. If you do so, **please clearly indicate which part is the extra one—I will assume it is part (c) if you mark none of them.**
- For each part of that problem, you must complete a short program. I have provided comments to describe what your program should do, as well as written some of the program for you.
- For `printf()` / `scanf()` calls, you must fill in the blanks provided. You may have to write additional code outside of these functions.
- You can solve each problem using only the variables that have been declared, but you may declare and use other variables if you want.

You will have 50 minutes to complete this exam.

Q1: Multiple choice	/ 20
Q2: Expressions/operators	/ 40
Q3: C input/output; conditional statements	/ 40
TOTAL SCORE	/ 100
EXTRA CREDIT	/ 10

1. (20 points, 5 points per part) ***Multiple choice***
For each of the multiple choice questions below, clearly indicate your response by circling or underlining the choice you think best answers the question.

a. How many iterations does the loop below execute?

```
int x = 1;
while (x <= 20) {
    x = x * 2;
}
```

   i.   2

  ii.   4

 iii.   5

 iv.   10

  v.   20

b. Given the short code sequence below:

```
int z = 1;
do {
    printf("z = %d\n", z);
    z = z * -2;
} while (z < 0);
```

What will this program print?

   i.   Nothing

  ii.   z = 1

 iii.   z = 1
       z = -2

 iv.   z = 1
       z = -2
       z = 4

1 (cont.)

c. Given the code sequence below:

```
int x;
scanf("%d", &x);
switch (x / 4) {
case 0:
   printf("Zero ");
   break;
case 2:
   printf("Two ");
default:
   printf("Default");
}
```

Which of the following possible input values will produce the output "Two Default"?

    A. 0
    B. 2
    C. 4
    D. 8
    E. 9

    i.   A and B

    ii.   Only B

    iii.   B and C

    iv.   Only D

    v.   D and E

d. Which of the following statements accurately reflect your opinion(s)? Circle all that apply (but please don't waste too much time on this question)!

    i.   "This course is moving too quickly."

    ii.   "This course is moving too slowly."

    iii.   "I appreciate the opportunity to take a 50 minute nap three times a week."

    iv.   "I hope the rest of the exam is this easy."

    v.   "The programs get easier after Program 3, right? Please???"

2. (40 points) *Expressions/operators*

For each short program shown below, list the output exactly as it will appear on the screen. Be sure to clearly indicate spaces between characters when necessary.

You may use the available space to show your work as well as the output; just be sure to clearly mark where you show the output so that I can easily recognize your final answer.

a. (12 points)

```
#define Q2PartA 7 / 2.0

void main() {
    int intA, intB, intC;
    double doubA;

    intA = Q2PartA + 8;
    intB = 15 - intA % 6;
    intC = 30 / (intA + intB);

    doubA = intB - 0.5;

    printf("Ints:\n%d %d %d", intA, intB, intC);
    printf("\nDouble: %lf", doubA);
}
```

2 (cont.)

b. (14 points)

```
void main() {
    double x = 10;
    double y = 4;
    double out1, out2, out3;

    out1 = x / y / x;
    out2 = y * (y + 1);
    out3 = 3.7 + 30 / x;

    printf("%5.2lf %+3.3lf\n", out1, out2);
    printf("%-6.0lf\n", out3);
}
```

c. (12 points)

```c
void main() {

    unsigned int w, x, y;
    w = 0xFF0000 | 0xAA;
    x = w ^ (0xBC << 8);
    y = w & 0xF00F0;

    printf("%#.8X\n", w);
    printf("%#.8X\n", x);
    printf("%#.8X\n", y);
}
```

3. (40 points, 20 per part) *C input/output; conditional statements*
For each part of this problem, you are given a short program to complete. **CHOOSE ANY TWO OF THE THREE PARTS** and fill in the spaces provided with appropriate code. **You may complete all three parts for up to 10 points of extra credit, but must clearly indicate which part is the extra one—I will assume it is part (c) if you mark none of them.**

a. Complete the program below so that it prompts for and reads two integers representing a time in 24-hour format ($0 \leq$ hours ≤ 23), then prints the time in 12-hour format:

Valid range of hours (in 24 hr format)	AM / PM (12 hr format)	Earliest time (24 hr → 12 hr)	Latest time (24 hr → 12 hr)
0 ≤ hours ≤ 11	AM	0:00 == 12:00 AM	11:59 == 11:59 AM
12 ≤ hours ≤ 23	PM	12:00 == 12:00 PM	23:59 == 11:59 PM

All `printf()` calls are shown; you must fill in the blanks so they produce the right output. Examples are shown below, with user input underlined:

```
Hrs/mins: 11 13          Hrs/mins: 14 5          Hrs/mins: 0 9
11:13 AM                 2:05 PM                 12:09 AM
```

```
void main() {
      int hrs, mins;        // Time inputs: hours and minutes

      // Prompt for and read time in 24 hour format
      printf("Hrs/mins: ");

      scanf("_____", _____);

      // To handle "12:00" cases (hrs == 0 or 12),
      //    add 12 to hrs if hrs is divisible by 12




      // Check if time is in morning and print accordingly



            printf("_____ AM\n", _____);

      // Otherwise, time must be in afternoon



            printf("_____ PM\n", _____);

}
```

7

3 (cont.)

b. Complete the program below so that it first prompts for and reads two lines of input. The first contains the lower and upper bounds of a range; the second contains two values to be tested. The program should then print the following output:

- If `in2` is outside of the range, print "`I2 out of range`".
- If both are in the range (including the end points), print "`Both in range`".
- If neither of the previous conditions is true, print "`Conditions false`".

Three sample program runs are shown below, with <u>user input underlined</u>:

```
 Range: 5 10            Range: 3.5 7.1          Range: 0 2.1
 Values: 1 11           Values: 5.9 3.5         Values: 5 1
 I2 out of range        Both in range           Conditions false
```

```c
void main() {
    double low, hi;          // Lower and upper bounds
    double in1, in2;         // Values to be checked

    // Prompt for and read range
    printf("Range: ");

    scanf("_____", _____);

    // Prompt for and read values
    printf("Values: ");

    scanf("_____", _____);

    // If in2 is out of range, print message below



        printf("I2 out of range\n");

    // If not, check if input1 is in range;
    //    if so, print message below



        printf("Both inputs in range\n");

    // In all other cases, print message below


        printf("Conditions false\n");
}
```

8

3 (cont.)

c. Complete the program below so that it prompts for and reads a single letter followed by two numbers. Depending on the character entered, your program should print the following information about these numbers, always showing one digit after the decimal point:

- 'A','M' → print the average in the form: `Avg = <result>`
- 'D' → print the difference `num1 – num2` in the form: `Diff = <result>`
- Any other character → print "`Invalid command`"

Note that all `printf()` calls are shown; you are responsible for filling in the blanks so that they produce the correct output.

Three sample program runs are shown below, with <u>user input underlined</u>:

```
Input: A 2 5            Input: L 0 4.27         Input: m 17 3
Avg = 3.5               Diff = -4.3             Invalid command
```

```
void main() {
    char cmd;              // Command input
    double num1, num2;    // Two numeric inputs

    // Prompt for and read input
    printf("Input: ");

    scanf("_____", _____);

    // Based on input, print avg., diff., or "Invalid command"




    printf("Avg = _____\n", _____);




    printf("Diff = _____\n", _____);




    printf("Invalid command\n");

}
```

9