

# 16.216: ECE Application Programming

Fall 2014

Exam 1

October 1, 2014

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 Question 3, you must complete a short program. I have provided comments to describe what your program should do and written some of the code.
  - Note that each program contains both lines that are partially written (for example, a `printf()` call in which you are responsible for filling in the format string and expressions) and blank spaces in which you must write additional code. **You must write all code required to make each function work as described—do not assume you can simply fill in the blank lines and get full credit.**
  - Each program is accompanied by one or more test cases. Each test case is an example of how the program should behave in one specific case—**it does not cover all possible results of using that function.**
- You can solve each part of Question 3 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: C input/output; operators	/ 40
Q3: Conditional statements	/ 40
<b>TOTAL SCORE</b>	<b>/ 100</b>
<b>EXTRA CREDIT</b>	<b>/ 10</b>

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 one choice you think best answers the question.

a. What is the output of the short code sequence below?

```
x = 1;
while ((x % 3) != 0) {
    printf("%d ", x);
    x = x + 4;
}
```

i. No output

ii. 1

iii. 1 5

iv. 1 5 9

v. 1 5 9 13

b. What is the output of the short code sequence below?

```
int x = 3;
int y = -3;
do {
    printf("* ");
    y = y + 1;
    x = -y + 2;
} while ((x > y) && (x > 2));
```

i. No output

ii. \*

iii. \* \*

iv. \* \* \*

v. \* \* \* \*

1 (continued)

c. Given the following code snippet:

```
int x = 100;
for (i = 1; i < 5; i++) {
    x = x - 10;
}
```

Which of the following choices can replace the `for` loop and produce the exact same value for `x`? Assume `x` is always initialized to 100.

- i. 

```
i = 0;
while (i < 8) {
    x += (-10);
    i += 2;
}
```
- ii. 

```
i = 5;
while (i >= 1) {
    x = x - 10;
    i--;
}
```
- iii. 

```
i = 0;
while (i < 5) {
    x = x - 10;
    i++;
}
```
- iv. 

```
i = x;
while (i > 60) {
    i = i - 10;
}
```

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've attended very few lectures, so I don't really know what the pace of the course is."
- iv. "I hope the rest of the exam is this easy."

2. (40 points) *C input/output; 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)

```
void main() {
    double d1, d2;
    int v1;
    int v2 = 16;

    d1 = v2 / 3;
    d2 = (v2 + 1.0) / 2;
    v1 = d1 * 2 + 7;
    v2 = v2 % v1;

    printf("%lf %lf ", d1, d2);
    printf("%d\n%d\n", v1, v2);
}
```

2 (continued)

b. (14 points)

```
void main() {  
    int in1;  
    float f1, f2;  
    double do1;  
  
    in1 = 20;  
    f1 = in1 / 40.0 + 3;  
    f2 = f1 + in1;  
    do1 = (f2 - f1) / (in1 - 15);  
    in1 = f1 + f2;  
  
    printf("%06d\n", in1);  
    printf("%-6.2f %+6.2f\n", f1, f2);  
    printf("%10.4lf\n", do1);  
}
```

2 (continued)

c. (14 points)

For this program, assume the user inputs the line below. The digit '1' is the first character the user types. The plus sign ('+') is followed by a single space character (' '), while the digits '4' and '5' are separated by two spaces.

You must determine how `scanf()` handles this input and then print the appropriate results. Note that the program may not read all characters on the input line.

```
110+ 0.1234  5
```

```
void main() {
    int ival;
    double dval;
    char ch1, ch2, ch3, ch4;

    scanf("%c%d%c %c %lf %c",
        &ch1, &ival, &ch2, &ch3, &dval, &ch4);

    printf("%d %.4lf\n", ival, dval);
    printf("%c%c%c%c\n", ch1, ch2, ch3, ch4);
}
```

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.**

Remember, you must write all code required to make each program work as described—**you cannot simply fill in the blank lines and get full credit.** Also, remember that each example only applies to one specific case—**it does not cover all possible results for that program.**

- a. The first three digits of a 5-digit zip code are based on the state in which the zip code is located. For example, all Massachusetts zip codes start with digits between 010 and 027, and all New Hampshire zip codes start with digits between 030 and 038.

Your program should read a zip code and determine if it is in one of these two states. The program should also check that the input is properly formatted as an integer, printing an error message if not. Three test runs of the program are shown below, with user input underlined.

Enter zip: <u>01854</u>	Enter zip: <u>06320</u>	Enter zip: <u>X6320</u>
01854 --> MA	06320 outside MA & NH	Input error

Note that you can read and print integers with leading zeroes, but you should not use leading zeroes for constant values in your program. For example, to check if the user entered the university's zip code, you would compare zip to 1854, not 01854. Note also that leading zeroes do not affect how you read an integer, but you will need extra formatting to print leading zeroes.

```
void main() {
    int zip;           // Input zip code
    int n;             // Test for input error

    // Prompt for and read zip code
    printf("Enter zip code: ");

    if (_____) // Test for input error
        printf("Input error\n");

    else if (_____) // MA zip codes
        printf("_____ --> MA\n", _____);

    else if (_____) // NH zip codes
        printf("_____ --> NH\n", _____);

    else
        printf("_____ outside MA & NH\n", _____);
}
```

3 (continued)

- b. Given three double-precision input values, this program should print the greatest of the inputs using a precision of 2. Two sample program runs are shown below (user input underlined):

Enter three values: 1.2 3.4 5.6

5.60 is max value

Enter three values: -7 3 1

3.00 is max value

```
void main() {
    double d1, d2, d3;           // Three inputs
    double max;                  // Maximum value

    // Prompt for and read three input values
    printf("Enter three values: ");

    scanf("_____", _____);

    // Use if/else statements to find maximum value. HINT: Use
    // nested if statements—compare two inputs (which will help
    // you eliminate one) with an if statement, then use an
    // if/else statement inside that block to choose one of the
    // other inputs. Handle the outer else case similarly.

    // Print maximum value

    printf("_____ is max value\n", _____);
}
```



3 (continued)

- c. This program prompts for and reads a double-precision value followed by a single character representing one of three prefixes: 'k', for kilo- ( $10^3 = 1$  thousand), 'M', for mega- ( $10^6 = 1$  million), or 'G', for giga- ( $10^9 = 1$  billion). Depending on the character, your program should multiply the input value by the appropriate amount and show the full value rounded to the nearest integer. If the user enters an invalid character, print an appropriate error message.

Two sample program runs are shown below, with user input underlined:

Enter value, char: <u>1.0543X</u>	Enter value, char: <u>77.9555 G</u>
Invalid character X	Final value: 77955500000

```
void main() {
    double val; // Input value
    char pref;  // Character representing prefix

    // Prompt for and read inputs
    printf("Enter value, char: ");

    scanf("_____", _____);

    // Determine appropriate multiplier and change val accordingly
    switch (_____) {

    case _____:

    case _____:

    case _____:

    _____: // Error case

        printf("Invalid character _____\n", _____);
        return;
    }

    // Print final result after multiplication
    printf("Final value: _____\n", _____);
}
```