

# 16.216: ECE Application Programming

Spring 2015

Exam 1  
February 23, 2015

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 if x is 10?

```
switch (x) {  
  case 10:  
    printf("A ");  
  case 20:  
    printf("B ");  
  default:  
    printf("C ");  
}
```

- i. A
- ii. B
- iii. C
- iv. A B
- v. A B C

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

```
int i = 0;  
while ((i * i) < 15) {  
  printf("%d ", i);  
  i = i + 2;  
}
```

- i. 0
- ii. 0 2
- iii. 0 4
- iv. 0 2 4
- v. 0 4 16

1 (continued)

c. Which of the following if statements will print "True" if x is 216?

- A. 

```
if (x != 216)
    printf("True");
```
- B. 

```
if ((x < 220) && (x > 210))
    printf("True");
```
- C. 

```
if ((x < 200) || (x > 240))
    printf("True");
```
- D. 

```
if (x)
    printf("True");
```

- i. Only A
- ii. Only B
- iii. A and C
- iv. B and D
- v. All of the above (A, B, C, and D)

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 didn't know we had lectures—I thought every day was a snow day."
- v. "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() {
    int v1;
    int v2 = 35;
    double d1, d2;

    v1 = v2 / 10;
    d1 = (v2 - 20.0) / 10;
    d2 = d1 + 2 * d1 - 7;
    v2 = d2 + d1;

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

2 (continued)

b. (14 points)

```
void main() {
    double d1, d2, d3;
    int x = 42;

    d1 = x / 8;
    d2 = (d1 + 8) / (d1 - 15);
    d3 = d1 + d2 + 0.0975;
    x = d3 - d2;

    printf("%d\n", x);
    printf("%.31f\n", d1);
    printf("%.01f\n", d2);
    printf("%.21f\n", d3);
}
```

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. There are two spaces (' ') between the '6' in 16.216 and the '2' in 2.23, and two spaces between the '3' in 2.23 and the '9' in 950.

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.

```
16.216  2.23  950
```

```
void main() {
    int ival1, ival2;
    double dval1, dval2;
    char ch1, ch2, ch3;

    scanf("%d%c%lf%c %lf %c %d",
          &ival1, &ch1, &dval1, &ch2,
          &dval2, &ch3, &ival2);

    printf("%d %d\n", ival1, ival2);
    printf("%.31f %.31f\n", dval1, dval2);
    printf("%c%c%c\n", ch1, ch2, ch3);
}
```

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. This program should first prompt for and read two pairs of numbers; each pair represents the lower and upper bounds of a range. The program should then do the following:
- If the ranges overlap, even if it's just at one endpoint each, print "Overlap".
  - If range 1 is lower (below low endpoint of range 2), print "Range 1 < Range 2".
  - If range 2 is lower, print "Range 1 > Range 2".

Three test runs of the program are shown below, with user input underlined.

Range 1: <u>1.5 3.5</u>	Range 1: <u>3.14 3.24</u>	Range 1: <u>16 216</u>
Range 2: <u>2.7 8.7</u>	Range 2: <u>3.25 9.25</u>	Range 2: <u>-2 15</u>
Overlap	Range 1 < Range 2	Range 1 > Range 2

```
int main() {
    double L1, H1;    // Endpoints of range 1
    double L2, H2;    // Endpoints of range 2

    // Prompt for and read ranges, value
    printf("Range 1: ");

    scanf("_____", _____);
    printf("Range 2: ");

    scanf("_____", _____);

    // Test relationship between ranges

    printf("Overlap\n");

    printf("Range 1 < Range 2\n");

    printf("Range 1 > Range 2\n");
}
```

3 (continued)

b. Your code will print the grade represented by a score between 70 and 99, given these rules:

- Scores between 70-79 are in the C range, 80-89 in the B range, and 90-99 in the A range.
- If the second digit of the score is between 0-2 (for example, 70-72), add a '-' to the letter grade; if that digit is between 8-9 (for example, 88-89), add a '+' to the letter grade.

Rather than just checking possible grade ranges, your program should split the score into two digits and evaluate each separately. (Hint:  $93 / 10 = 9$ , with a remainder of 3.)

Three sample program runs are shown below (user input underlined):

```
Enter score: 83           Enter score: 79           Enter score: 91
Grade: B                   Grade: C+                   Grade: A-
```

```
void main() {
    int score;           // Score entered by user
    int hi, lo;         // High and low score digits

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

    scanf("_____", _____);

    // Separate digits (example: if score = 93, hi = 9 and lo = 3)

    // Print letter grade (write code around printf() calls)

        printf("Grade: A");

        printf("Grade: B");

        printf("Grade: C");

    // Print second part of grade (+ or -, if necessary)
    if (_____ )
        printf("+");
    else if (_____ )
        printf("-");
}
```



3 (continued)

c. This program takes an integer between 1 and 23 representing a day in February thus far and prints an appropriate description. The notes below will help you identify the non-school days:

- All Saturday dates are divisible by 7.
- February 2, 3, 9, and 10 were all snow days.
- February 16 was Presidents' Day.

```
void main() {
    int day;           // Day within February

    // Prompt for and read day
    printf("Enter day in February: ");
    scanf("_____", _____);

    // Error: invalid date (not between 1 and 23)
    if (_____)
        printf("Invalid date\n");

    // Weekend—check if day was Saturday or Sunday
    else if (_____)
        printf("Weekend\n");

    // Other cases—write code around print statements that checks
    // specific dates described above
    else {

        printf("Snow day\n");

        printf("Presidents' Day\n");

        printf("School day\n");

    }
}
```