

16.216: ECE Application Programming

Summer 2012

Exam 1
July 24, 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 by writing all input and output statements (except prompts). I have provided comments to describe what your input/output statements should do, as well as written the rest of the program for you.
- 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 1 hour to complete this exam.

Q1: Multiple choice	/ 20
Q2: Expressions and operators	/ 40
Q3: C input/output	/ 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. Say you have the following code snippet:

```
double v1, v2;
scanf("%lf %lf", &v1, &v2);
if ((v1 < v2) && (v1 > 5.0))
    printf("Condition true\n");
```

Which of the possible lines of input below will cause this code to print `Condition true`?

- i. 0.75 3.75
- ii. 45 5.4
- iii. 5.0 6.7
- iv. 5.1 6.7
- v. `Condition true`

b. Say you have the following conditional statement (assume all variables are of type `int`):

```
if (b + c == 3) {
    a = 5;
}
else if (b < c) {
    a = 17;
}
else {
    a = 34;
}
```

Which values of `b` and `c` shown below would cause `a` to be set to 17 when the code above is executed?

- i. `b == 2, c == 2`
- ii. `b == -2, c == 5`
- iii. `b == 2, c == 5`
- iv. `b == 12345, c == 0`

1 (cont.)

c. Say you have the following code snippet:

```
scanf("%d", &x);
switch(x % 3) {
case 0:
    printf("Result 0 ");
case 1:
    printf("Result 1 ");
    break;
default:
    printf("Default");
}
```

If you run the program containing this code twice, and input the values 9 and 8, respectively, what will the outputs be?

- i. First run (x == 9): Result 0
Second run (x == 8): Default
- ii. First run (x == 9): Result 0 Result 1
Second run (x == 8): Default
- iii. First run (x == 9): Result 0 Result 1 Default
Second run (x == 8): Default
- iv. First run (x == 9): Default
Second run (x == 8): Default
- v. First run (x == 9): Default
Second run (x == 8): Result 0 Result 1

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. "I'd rather be outside than stuck in class."
- ii. "This course is moving too quickly."
- iii. "This course is moving too slowly."
- iv. "I hope the rest of the exam is this easy."
- v. "We don't really have to show up for the 'extra' Friday classes, do we?"

2. (40 points) *Expressions and 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 3.0 / 2

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

    intA = 7 + Q2PartA;
    intB = 2 + intA / 3 * 2;
    intC = 3 * (intA - intB);

    doubA = intA + Q2PartA;

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

2 (cont.)

b. (14 points)

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

    x = 7;
    d1 = x + 3.5 / 5;
    d2 = (d1 + 2.3) / 4 + 0.008;
    d3 = x / 2.0 + 1.1;
    x = d2;

    printf("d1 = %-10.3f\n", d1);
    printf("d2 = %*.*f\n", x, x, d2);
    printf("d3 = %+0f\n", d3);
}
```

c. (14 points)

```
void main() {  
    unsigned int H1, H2, H3;  
  
    H1 = 0xFF0 & 0x123;  
    H2 = ~(H1 << 8);  
    H3 = H1 ^ 0x0000FF00;  
  
    printf("H1: %x\n", H1);  
    printf("H2: %#x\n", H2);  
    printf("H3: %#.8x\n", H3);  
}
```

3. (40 points, 20 per part) C input/output

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 space 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 this program so that it prompts for and reads three hexadecimal values (*in1*, *in2*, *in3*) and a single integer (*x*), then prints the following on three separate lines:
- First line: show *in1* both in decimal and in hexadecimal, with the hexadecimal output showing all 32 bits with a leading 0x and extra zeroes if necessary.
 - Second line: show *in2* multiplied by 2^x , in hexadecimal. This output should have only a leading 0x, with no additional formatting.
 - Third line: show the lowest 12 bits of *in3*, with a leading 0x and extra zeroes if necessary.

Examples (with user input underlined):

```
3 hex, 1 int: A 80 11F3 2  
10 0x0000000A  
0x200  
0x1F3
```

```
3 hex, 1 int: 0100 0xF 0x3 10  
256 0x00000100  
0x3C00  
0x003
```

```
void main() {  
    unsigned int in1, in2, in3;    // Hexadecimal inputs  
    int x;                          // Integer input  
  
    // Prompt for and read three hexadecimal values and one int  
    printf("3 hex, 1 int: ");  
  
    // Print in1 in decimal and hexadecimal (show all 32 bits)  
  
    // Print in2 multiplied by 2x  
  
    // Print only the lowest 12 bits of in3  
  
}
```

3 (cont.)

b. Complete this program so that it prompts the user to enter four input values representing a baseball player's statistics—hits (H), walks (BB), total bases (TB), and at bats (AB), then calculates and prints the following values, each on its own line:

- Batting average (AVG): the fraction of at bats in which the player gets a hit
- On-base percentage (OBP): the fraction of plate appearances ($PA = AB + BB$) in which the player reaches base by getting either a walk or a hit.
- Slugging percentage (SLG): the ratio of total bases to at bats.
- OPS: the sum of OBP and SLG.

Each value should be printed using 3 decimal places. Examples are shown below, with user input underlined:

```
Enter H / BB / TB / AB: 100 50 180 450
AVG: 0.222
OBP: 0.300
SLG: 0.400
OPS: 0.700
```

```
void main() {
    double H, BB, TB, AB, PA;        // Stats described above

    // Prompt for and read hits, walks, total bases, at bats
    printf("Enter H / BB / TB / AB: ");

    // Calculate PA

    // Print AVG, OBP, SLG, and OPS

}
```

3 (cont.)

- c. Complete this program so that it prompts the user to enter a student ID number followed by three more non-negative integers, reads those values, and prints the following on one line:
- The student ID number, as an 8-digit value.
 - If the ID has less than 8 digits, leading zeroes should fill the remaining places.
 - Each of the three integers, with at least one space between each
 - The integers should always line up the same way, regardless of the number of digits
 - Each integer has a minimum value of 0 and a maximum value of 100.
 - The average of the three integers, always shown with 2 digits after the decimal point

```
Enter ID and 3 ints: 12345678 50 60 70
12345678 50 60 70 60.00
```

```
Enter ID and 3 ints: 33 5 100 35
00000033 5 100 35 46.67
```

```
void main() {
    unsigned int ID;           // Student ID
    unsigned int v1, v2, v3;   // Three integer inputs

    // Prompt for and read ID number + 3 integers
    printf("Enter ID and 3 ints: ");

    /* Print 8-digit ID, 3 integer values (always aligned the
       same way), & average (with 2 digits after decimal pt) */

}
```