

EECE.2160: ECE Application Programming

Fall 2016

Exam 1 Solution

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?

```
int i = 1;
int j = 15;

while (i < j) {
    i = i + i;
    j = j - 3;
    printf("%d %d > ", i, j);
}
```

i. 1 15 >

ii. 2 12 >

iii. 1 15 > 2 12 > 4 9 >

iv. 2 12 > 4 9 > 8 6 >

v. No output—loop body never executes

1 (continued)

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

```
int v = 9;

do {
    printf("%d ", v);
    v = v / 2;
} while ((v % 2 == 0) && (v > 0));
```

- i. No output—loop body never executes
- ii. 9
- iii. 9 4
- iv. 9 4 2**
- v. 9 4 2 1

c. Given the code sequence below:

```
int x, y;  
scanf("%d %d", &x, &y);  
switch (x + y) {  
case 5:  
    printf("Right ");  
case 10:  
case 15:  
    printf("Wrong ");  
default:  
    printf("Answer");  
}
```

Which of the following possible input pairs will produce the exact output “Wrong Answer ”?

- A. 2 3
- B. 5 5
- C. 5 -10
- D. -4 -6
- E. -10 25

- i. Only A
- ii. Only B
- iii. A and C
- iv. B and E**
- v. A, 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 hope the rest of the exam is as easy as this question.”

All of the above are “correct.”

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)

```
int main() {
    int iv1;
    int iv2 = 45;
    double dv1, dv2;

    dv1 = 5.0 + iv2 / 10;      dv1 = 5.0 + 45 / 10
                                = 5.0 + 4 (int division) = 9.0

    iv1 = iv2 % 10;           iv1 = 45 % 10 = 5

    dv2 = dv1 / (iv1 - 3);    dv2 = 9.0 / (5 - 3)
                                = 9.0 / 2 = 4.5

    iv2 = dv1 * dv2;          iv2 = 9.0 * 4.5 = 40.5 = 40
                                (truncated to int)

    printf("%d %d ", iv1, iv2);
    printf("%lf\n%lf\n", dv1, dv2);

    return 0;
}
```

OUTPUT:
5 40 9.000000
4.500000

2 (continued)
b. (14 points)

```
int main() {
    double d1, d2;
    double d3 = 15.75;
    int x = 4;

    d1 = 5.0 / x;
    x = d1 + d3;

    d2 = d3 * 100 - 70.2;

    d3 = d3 - 10;

    printf("%d\n", x);
    printf("%.3lf\n", d1);
    printf("%.0lf\n", d2);
    printf("%.1lf\n", d3);    // Print with precision of 1

    return 0;
}
```

d1 = 5.0 / 4 = 1.25
x = 1.25 + 15.75 = 17 (truncated to int after addition done)
*d2 = 15.75 * 100 - 70.2 = 1575 - 70.2 = 1504.8*
d3 = 15.75 - 10 = 5.75

OUTPUT:

17
1.250
1505 *← d2 rounded to nearest integer when printed*
5.8 *← d3 rounded to 1 decimal place when printed*

2 (continued)
c. (14 points)

For this program, assume the user inputs the line below. The digit '4' is the first character the user types. There is one space (' ') between the '0' in 4.50 and the '\$', one space between the last '9' in 9.99 and the '2' in 2016, and one space between the '6' in 2016 and the '9' in 96.

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.

4.50 \$9.99 2016 96

```
int main() {
    int ival1, ival2;
    double dval1, dval2;
    char c1, c2, c3, c4;

    scanf("%d%c%d %c%lf%c%lf %c",
          &ival1, &c1, &ival2, &c2,
          &dval1, &c3, &dval2, &c4);

    printf("%d %d\n", ival1, ival2);
    printf("%.2lf %.2lf\n", dval1, dval2);
    printf("%c%c%c%c\n", c1, c2, c3, c4);

    return 0;
}
```

Solution:

`ival1 = 4` (only integer part is read)
`c1 = '.'` (the first character after the 4)
`ival2 = 50` (next integer after the decimal point)
`c2 = '$'` (first non-space character after 50)
`dval1 = 9.99` (next real number after \$)
`c3 = ' '` (first character after 9.99 is a space)
`dval2 = 2016` (next real number after space)
`c4 = '9'` (first non-space character after 2016)

OUTPUT:

4 50
9.99 2016.00
.\$ 9

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 read three numbers: the mean and standard deviation of a normal distribution, and a number (num) to be tested to see where it falls in that distribution. The program should test three conditions and print an appropriate message:
- The number falls within 1 standard deviation of the mean (for example, if mean = 50 and sd = 5, the number is in the range $45 \leq \text{num} \leq 55$)
 - The number is outside the first range but within 2 standard deviations of the mean (if mean = 50 and sd = 5, num is either in the range 40-45 or in the range 55-60).
 - The number is greater than 3 standard deviations away from the mean (if mean = 50 and sd = 5, num is less than 35 or greater than 65).

```
void main() {
    double num;           // Input value
    double mean;          // Average
    double sd;            // Standard deviation

    // Prompt for and read number, mean, and standard deviation
    printf("Enter #, mean, SD: ");

    scanf("%lf %lf %lf", &num, &mean, &sd);

    // Test for ranges described above
    if (num >= mean - sd && num <= mean + sd)
        printf("Within 1 SD\n");
    else if (num >= mean - 2 * sd && num <= mean + 2 * sd)
        printf("Within 2 SD\n");
    else if (num < mean - 3 * sd || num > mean + 3 * sd)
        printf("Outside 3 SD\n"); }
}
```

3 (continued)

b. Assume we have a 4x4 grid, with rows A, B, C, and D, and columns 1, 2, 3, 4. A row/column pair—for example, A2 or D1—represents each entry in that grid. This program prompts for and reads four such pairs, with a space before each row number, and tests the following conditions:

- If all four entries are in the same row, print "Row match"
- If all four entries are in the same column, print "Col match"
- Otherwise, print "No match"

You may assume no duplicate entries are entered.

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

Rows/cols: A1 A3 A4 A2
Row match

Rows/cols: B3 D3 C2 A3
No match

Students had to write bold, underlined, italicized code

```
void main() {
    char r1, r2, r3, r4;           // Rows (between A-D)
    int c1, c2, c3, c4;           // Columns (between 1-4)

    // Prompt for and read row column pairs
    printf("Rows/cols: ");

    scanf(" %c%d %c%d %c%d %c%d",
          &r1, &c1, &r2, &c2, &r3, &c3, &r4, &c4);

    // Test conditions and print appropriate messages
    if ((r1 == r2) && (r2 == r3) && (r3 == r4))
        printf("Row match\n");

    else if ((c1 == c2) && (c2 == c3) && (c3 == c4))
        printf("Col match\n");

    else
        printf("No match\n");
}
```


3 (continued)

c. This program prompts the user to enter a phone number in the form xxx-yyy-zzzz. xxx is the area code, yyy is the exchange, and zzzz the last four digits. Your program should test the area code and print the name of a major city in that area code, or “Other” if the area code matches none of the following:

- Area codes 978 and 351 cover the region containing Lowell.
- Area codes 617 and 857 cover the region containing Boston.
- Area code 413 covers the region containing Springfield.

The program should then print the remainder of the phone number. Two sample program runs are shown below, with the user input underlined:

Enter number: 978-934-3618
Lowell
Number: 934-3618

Enter number: 123-456-7890
Other
Number: 456-7890

```
void main() {
    int area, exch, last4;           // Area code, exchange, and last
                                    // 4 digits of the phone number
    char d1, d2;                    // Characters to read dashes

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

    scanf("%d%c%d%c%d", &area, &d1, &exch, &d2, &last4);

    // Test area code and print city name, then rest of number
    switch(area) {
        case 978: case 351:
            printf("Lowell\n");
            break;
        case 617: case 857:
            printf("Boston\n");
            break;
        case 413:
            printf("Springfield\n");
            break;
        default:
            printf("Other\n");
    }
    printf("Number: %d-%d", exch, last4);
}
```