

EECE.2160: ECE Application Programming

Spring 2018

Exam 1

February 23, 2018

Name: _____

Lecture time (circle 1): 8-8:50 (Sec. 201) 12-12:50 (Sec. 202)

For this exam, you may use only one 8.5" x 11" double-sided page of notes. All electronic devices (e.g., calculators, cell phones) are prohibited. Please turn off your cell phone ringer prior to the start of the exam to avoid distracting other students.

The exam contains 3 questions for a total of 100 points, plus a 10 point extra credit question. 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:

- Questions 1c and 2b require you to complete short programs. We have provided comments to describe what each 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 missing the format string and expressions) and blank spaces in which you must write additional code. **You must write all code required to make each program work as described—do not simply fill in the blank lines.**
 - Each test case is an example of how the program should behave in one specific case—**it does not cover all possible results of running that program.**
 - You can solve each of these questions using only the variables that have been declared, but you may declare and use other variables if you want.
- Carefully read the multiple choice problems. Questions 3a and 3b have exactly one correct answer, while Question 3c requires you to choose two answers.
- **You cannot earn any extra credit without partial solutions to all parts of Questions 1, 2, and 3.** In other words, don't try the extra credit until you've tried to solve every other question on the exam.

You will have 50 minutes to complete this exam.

Q1: C input/output; operators	/ 46
Q2: Conditional statements	/ 34
Q3: While and do-while loops	/ 20
TOTAL SCORE	/ 100
Q4: EXTRA CREDIT	/ 10

1. (46 points) *C input/output; operators*

- a. (13 points) Show the output of the short program below 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.

```
int main () {
    int i;
    double d1 = 9.0;
    double d2, d3;

    i = 4.5 + d1 / 2;
    d2 = d1 / 100.0;
    d3 = i + 0.21598;
    d1 = d1 / 10;

    printf("%d\n", i);
    printf("%.01f\n%.31f", d1, d2);
    printf(" %.41f\n", d3);           // Note: 1 space before '%'

    return 0;
}
```

1 (continued)

- b. (13 points) For this program, assume the user inputs the line below. The digit '2' in 2+1.60 is the first character the user types. There is exactly one space (' ') between 2+1.60 and 20.18.

You must determine how `scanf()` handles this input and then print the appropriate results, exactly as they would be shown on the screen. The program may not read all characters on the input line, but `scanf()` will read something into all seven variables declared in the program.

```
2+1.60 20.18
```

```
int main () {
    int int1, int2;
    double d1, d2;
    char ch1, ch2, ch3, ch4;

    scanf("%lf %c%d %c %d%c%c %lf",
          &d1, &ch1, &int1, &ch2,
          &int2, &ch3, &ch4, &d2);

    printf("%d %d\n", int1, int2);
    printf("%.2lf %.2lf\n", d1, d2);
    printf("%c%c%c%c\n", ch1, ch2, ch3, ch4);

    return 0;
}
```

1 (continued)

- a. (20 points) Complete this program, which calculates a student's overall GPA after 4 semesters, given the GPA and credits per semester. The program prompts for and reads the four GPAs and credit counts, then calculates and prints the appropriate values, as in the example below (input is underlined):

```
Enter GPAs: 3.5 3.2 2.7 3.8
Enter credits: 14 17 18 15
Total credits: 64
Overall GPA: 3.27
```

← **NOTE: GPA printed using 2 decimal places**

The overall GPA is based on a weighted average. For example, after 2 semesters, a student who earned a 3.5 GPA while taking 12 credits and a 3.0 GPA while taking 15 credits would have a GPA of $(3.5 * 12 + 3.0 * 15) / (12 + 15) = 3.22$.

```
void main() {
    double G1, G2, G3, G4;        // Grade point averages
    int C1, C2, C3, C4;         // Credits per semester
    int total;                   // Overall total credits

    // Prompt for and read GPAs and credit counts
    printf("Enter GPAs: ");

    scanf(_____, _____);
    printf("Enter credits: ");

    scanf(_____, _____);

    // Calculate and print total credits and overall GPA

    printf("Total credits: _____\n", _____);

    printf("Overall GPA: _____\n",
           _____);
}
```

2. (34 points) ***Conditional statements***

- a. (14 points) For the short program shown below, the first line of output (the prompt "Enter val, 2 sets of endpoints: ") and the user input (5.6 1.2 3.4 7.8 9.0) is listed at the bottom of the page.

Complete the rest of the output for this program, given those input values.

```
int main() {
    double testval;
    double r1lo, r1hi;
    double r2lo, r2hi;

    printf("Enter val, 2 sets of endpoints: ");
    scanf("%lf %lf %lf %lf %lf",
          &testval, &r1lo, &r1hi, &r2lo, &r2hi);

    if (testval <= r1hi && r1lo <= testval)
        printf("R1\n");
    else {
        if (r2lo > testval)
            printf("Below R2\n");
        if (r2hi < testval)
            printf("Above R2\n");
        else
            printf("In R2?\n");
    }

    if (r1lo <= r2hi && r2lo <= r1hi)
        printf("Overlap\n");
    else
        printf("No overlap\n");
    return 0;
}
```

OUTPUT (the first line is given; write the remaining line(s)):

Enter val, 2 sets of endpoints: 5.6 1.2 3.4 7.8 9.0

2 (continued)

b. (20 points) Complete this program, which implements two simple operations on a pair of integers. Your program should prompt for and read the operator and integers, then check for one of the conditions below, printing the appropriate output:

- If the operator is an uppercase or lowercase 'A', print the average of the integers.
- If the operator is an uppercase or lowercase 'M', print the higher (max) of the integers.
- In all other cases, print "Error".

All numeric outputs should be printed with two digits after the decimal point. Three test cases are shown below, with user input underlined.

Enter input: A 5 10 Enter input: m 6 2.1 Enter input: x 3 5
Avg = 7.50 Max = 6.00 Error

```
void main()
    char op;                      // Operator
    double v1, v2;                // Values to operate on
    double max;                  // Maximum value for 'M'/'m' case

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

    scanf("_____ ", _____);

    // Evaluate operator

    // Average

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

    // Max value

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

    // Invalid op

    printf("Error\n");
}
```

3. (20 points, 5 points each) ***While and do-while loops***

For questions 3a and 3b, circle or underline the one choice you think best answers the question.

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

```
int i = 4;
int j = 2;
while (i != j) {
    i = i + 4;
    j = j * 2;
    printf("%d %d ", i, j);
}
```

- i. 4 2 8 4 12 8 16 16
- ii. 4 2 8 4 12 8
- iii. 8 4 12 8 16 16
- iv. 8 4 12 8
- v. The loop prints nothing because the loop condition is initially false

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

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

- i. +
- ii. + +
- iii. + + +
- iv. + + + +
- v. + + + + +

3 (continued)

c. Which pair of loops below produce the exact same output? **Circle TWO choices to answer this question.**

i.

```
int a = 1;
while ((12 % a) == 0) {
    printf("%d ", a);
    a = a * 2;
}
```

ii.

```
int b = 1;
do {
    printf("%d ", b);
    b = b * 2;
} while (b > 4);
```

iii.

```
int c = 4;
do {
    printf("%d ", 5 - c);
    c = c / 2;
} while (c > 0);
```

iv.

```
int d = 1;
while (d < 5) {
    printf("%d ", d);
    d = d + 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 next exam is as easy as this question."

4. (10 points) **EXTRA CREDIT**

REMEMBER, YOU CANNOT GET EXTRA CREDIT WITHOUT WRITING AT LEAST PARTIAL SOLUTIONS FOR ALL OTHER PROBLEMS ON THE EXAM.

Complete the program below, which reads an integer, `val`, copies its original value so it may be reprinted at the end, and then determines the number of digits in that integer. The variable `n` should hold the number of digits in `val` when the program is done. For example, if `val = 5`, `n = 1`; if `val = 1033`, `n = 4`.

The number of digits in `val` can be found by repeatedly dividing `val` until the result is 0; the number of steps required to reach that point is the number of digits in `val`. For example, if `val = 16216`, the program goes through the sequence below to determine `val` has 5 digits:

16216 → 1621 → 162 → 16 → 1 → 0 (each arrow represents one step)

```
int main() {
    int val, valCopy;    // Input value and its copy
    int n;              // Number of digits in val

    printf("Enter number: "); // Prompt for and read input value
    scanf("%d", &val);

    // COMPLETE PROGRAM AS DESCRIBED ABOVE
```

```
    // Print final results
    printf("Value: %d, # digits: %d\n", valCopy, n);
    return 0;
}
```