EECE.2160: ECE Application Programming

Fall 2017

Exam 1

October 4, 2017

Name: ____

Lecture time (circle 1): *8-8:50 (Sec. 201)* *12-12:50 (Sec. 203)* *1-1:50 (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. If you have a cell phone, please turn off your 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 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 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.
- To ensure you do not ignore the main part of the exam in favor of the extra credit question, **you cannot earn any extra credit without writing at least partial solutions for all parts of Questions 1, 2, and 3.** In other words, don't try the extra credit question until you've attempted 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 () {
    double d1, d2, d3;
    int x = 4;
    d1 = x % 3 / 4.0;
    d2 = d1 * (x + 3);
    d3 = d2 - d1 * 2;
    x = d2 + d3;

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

    return 0;
}
```

1 (continued)

b. (13 points) For this program, assume the user inputs the line below. The digit `'2'` in `2160` is the first character the user types. There is one space (`' '`) between `2160` and `8.00`, and there is one space between `8.00` and `-10.20`.

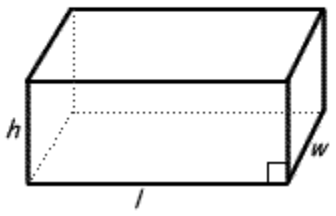
You must determine how `scanf()` handles this input and then print the appropriate results, exactly as they would be shown on the screen. The program may not read all characters on the input line, but `scanf()` will read something into all seven variables declared in the program.

```
      2160 8.00 -10.20

int main () {
    int i1, i2;
    double d1, d2;
    char ch1, ch2, ch3;

    scanf("%c %d%d%lf %c %c%lf",
       &ch1, &i1, &i2, &d1, &ch2, &ch3, &d2);

    printf("%d %d\n", i1, i2);
    printf("%.2lf %.2lf\n", d1, d2);
    printf("%c%c%c\n", ch1, ch2, ch3);

    return 0;
}
```

1 (continued)

c. (20 points) Complete this program, which reads the dimensions of a rectangular prism (a box with 6 rectangular sides, as shown on the right) with length L, width W, and height H, reprints the dimensions, and calculates and prints the volume and surface area of the prism. Note that:

- The volume of a rectangular prism is the product of all three dimensions.
- The surface area is the sum of the areas of all 6 sides.



The formatting of your output should exactly match the formatting of the two test cases below, in which user input is underlined:

```
Enter L, W, H: 2 3 4                 Enter L, W, H: 1.2 3.4 5.6
Length: 2.00                         Length: 1.20
Width: 3.00                          Width: 3.40
Height: 4.00                         Height: 5.60
Volume: 24.0000                      Volume: 22.8480
Surface area: 52.0000                Surface area: 59.6800
```

```
int main() {
    double L, W, H;       // Length, width, and height of prism

    // Prompt for and read dimensions
    printf("Enter L, W, H: ");

    scanf("_________________________",_________________________);

    // Reprint dimensions

    printf("Length: _____________\n", _________________________);

    printf("Width: _____________\n", _________________________);

    printf("Height: _____________\n", _________________________);

    // Calculate and print volume and surface area

    printf("Volume: _____________\n", _________________________);

    printf("Surface area: _____________\n",

          ____________________________________________________);

    return 0;
}
```

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

a. (14 points) For the short program shown below, the first line of output (the prompt `"Enter values: "`) and the user input (`3 1`) is listed at the bottom of the page. Complete the rest of the output for this program, given those input values.

```
int main() {
   int in1, in2;

   printf("Enter values: ");
   scanf("%d %d", &in1, &in2);

   switch (in1 + in2) {
      case 0:
         if (in1 > in2)
            printf("%d > %d\n", in1, in2);
         else
            printf("%d <= %d\n", in1, in2);
         break;

      case 2: case 4: case 6:
         if (in1 % 2 == 1)
            printf("%d is odd\n", in1);
         if (in2 % 2 == 0)
            printf("%d is even\n", in2);
         else
            printf("%d is odder still\n", in2);

      default:
         printf("%d isn't 4, is it?\n", in1 + in2);
   }
   return 0;
}
```

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

```
Enter values: 3 1
```

2 (continued)

b. (20 points) Complete this program, which implements a simple version of the game Yahtzee using only 3 dice. Your program should prompt for and read the 3 dice values, then check for one of the conditions below. Assume dice are entered in order from lowest to highest value.

- If all three dice match, print `"3 of a kind"`.
- If only two of the three dice match, print `"2 of a kind"`.
- If all three dice hold consecutive values (i.e., 3 4 5 or 1 2 3), print `"Straight"`.
- In all other cases, print `"Chance"`.

Three test cases are shown below, with user input underlined.

```
Enter 3 dice: 4 4 6    Enter 3 dice: 4 5 6    Enter 3 dice: 1 3 5
2 of a kind            Straight               Chance
```

```
void main() {
  int d1, d2, d3;        // Values of 3 dice

  // Prompt for and read dice values
  printf("Enter 3 dice: ");
  scanf("____________________", ______________________________);

  // Check for possible results


    printf("3 of a kind\n"); // All 3 dice match



    printf("2 of a kind\n"); // 2 out of 3 dice match



    printf("Straight\n");    // All 3 dice are consecutive



    printf("Chance\n");      // All other cases
}
```

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

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 = 8;
int j = -1;
while (j < i) {
   i = i - 2;
   printf("%d %d ", i, j);
   j = j + 1;
}
```

i. `8 0 6 1 4 2`

ii. `8 -1 6 0 4 1`

iii. `6 0 4 1 2 2`

iv. `6 -1 4 0 2 1`

v. `8 6 7 5 3 0 9`

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

```
int i = 17;
do {
   printf("%d ", i);
   i = i / 2;
} while (i % 2 == 0);
```

i. `17`

ii. `17 8 4 2`

iii. `8 4 2 1`

iv. `17 8 4 2 1`

v. This code sequence produces no output

3 (continued)

c. Which of the loops below will print the output: `0 1 2`?

i.
```
int i = 0;
while (i < 3) {
  printf("%d ", i);
}
```

ii.
```
int j = 2;
do {
   printf("%d ", j);
   j = j - 1;
} while (j >= 0);
```

iii.
```
int k = 9;
while (k - 1) {
   printf("%d ", k % 3);
   k = k / 2;
}
```

iv.
```
int m = 0;
do {
   printf("%d ", m);
   m = m + 1;
} while (m > 2);
```

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.

Write a short code sequence to read an expression of arbitrary length using integers and + and − operators, and calculate and print the result of the expression. For example, if the user enters:

```
1+2-3+4+5-6
```

your code would read that expression and print

```
Result = 3
```

You may assume:

- The shortest possible expression is a single number with no + or −. So, for example, if the user enters `15`, your code simply prints: `Result = 15`
- The only characters the user will input are digits, plus signs (`'+'`), minus signs (`'-'`) and a single newline character (`'\n'`), which indicates the end of the expression. No other characters—including spaces—will be entered, so don't account for them.
- The user always enters a valid expression, so don't account for any error cases.

If you would like additional space to write your solution, use the next page.

4 (continued) ***<u>EXTRA SPACE TO SOLVE EXTRA CREDIT PROBLEM</u>***