

# EECE.3220: Data Structures

Spring 2019

Exam 1

February 25, 2019

Name: \_\_\_\_\_

For this exam, you may use two 8.5" x 11" double-sided pages of notes. All electronic devices (e.g., calculators, cell phones) are prohibited. If you have a cell phone, please turn off your ringer before the start of the exam to avoid distracting other students.

The exam contains 6 sections for a total of 100 points, as well as a 10 point extra credit question. Please answer all 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:

- In Section 2, you must complete a short function. We have provided comments to describe what your function should do and written some of the code.
  - The function contains both lines that are partially written and blank spaces in which you must write additional code. **You must write all code required to make the function work as described—do not simply fill in the blank lines.**
  - You can complete the function using only the variables that have been declared, but you may declare and use other variables if you want.
- Each question in Section 6 is a multiple choice question that has at least one correct answer, but may have more than one correct answer.

You will have 2 hours to complete this exam.

S1: C++ input/output	/ 24
S2: Structures	/ 14
S3: Functions	/ 12
S4: C++ strings	/ 14
S5: Algorithmic complexity	/ 20
S6: Classes	/ 16
<b>TOTAL SCORE</b>	<b>/ 100</b>
<b>S7: EXTRA CREDIT</b>	<b>/ 10</b>

1. (24 points) C++ input/output

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 ival = 3220;
    double dval = 2.25;

    cout << ival << dval;
    cout << showpoint << ", ival = " << ival
        << ", dval = " << dval;
    cout << fixed;
    cout << "\nival = " << ival << ", dval = " << dval;
    cout << setprecision(1) << endl;
    cout << "ival = " << ival << ", dval = " << dval << endl;
    return 0;
}
```

1 (continued)  
b. (12 points)

For this program, assume the user inputs the two lines below. The letter 'E' in EECE is the first character the user types. There is one space ( ' ') between EECE . 3220 and E1, and no spaces on the second line of input. Assume each line ends with a newline character ( '\n ' ).

You must determine how the program handles this input and then print the appropriate results. Note that the program may not read all characters on the input lines, but no input statement fails to read data—an appropriate value is assigned to every variable.

```
EECE.3220 E1
2/25/2019
```

```
int main() {
    int i1, i2;
    double d1, d2;
    char c1, c2, c3;
    char buf[10];

    cin.ignore(4);
    cin >> d1 >> c1 >> i1;
    cin.get(c2);
    cin >> i2 >> c3 >> d2;
    cin.getline(buf, 10);

    cout << i1 << ' ' << i2 << endl;
    cout << d1 << ' ' << d2 << endl;
    cout << c1 << c2 << c3 << endl;
    cout << buf << endl;
    return 0;
}
```

2. (14 points) Structures

Complete the function below (the Box structure is defined on the extra sheet):

```
int maxVol(Box list[], int n);
```

This function takes as arguments an array of Box structures, `list`, as well as the number of structures in the list, `n`. The function returns the index of the structure representing the box with the greatest volume. For example, given:

```
Box arr[4] = { {1, 3, 5}, {9, 9, 9}, {4, 3, 2}, {2, 2, 2} };
```

the function call `maxVol(arr, 4)` would return 1, as `arr[1]` has the greatest volume (729, which is greater than the volumes of `arr[0]` (volume 15), `arr[2]` (24), and `arr[3]` (8)).

```
int maxVol(Box list[], int n) {
    int i;           // Loop index
    int maxI;       // Index of largest prism
    double maxVol;  // Volume of largest prism

    // Initialize variables as needed

    // Go through list; update max variables if larger box found
    for ( _____ ) {

        if ( _____ ) {

        }

    }

    // Return index of box with greatest volume
    return _____;
}
```

3. (12 points) **Functions**

For the 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.

```
void f1(int v1, int &v2) {
    int temp = v1;
    v1 = v2;
    v2 = temp;
}

void f2(int *v3, int v4) {
    int temp = *v3;
    *v3 = v4;
    v4 = temp;
}

void f3(int *v5, int &v6) {
    int temp = *v5;
    *v5 = v6;
    v6 = temp;
}

int main() {
    int x = 10;
    int y = 20;
    int z = 30;

    cout << x << " " << y << " " << z << endl;
    f1(x, y);
    f2(&y, z);
    f3(&z, x);
    cout << x << " " << y << " " << z << endl;

    return 0;
}
```

4. (14 points) Strings

For the 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.

```
int main() {
    string s1("question"), s2("act"), s3;
    unsigned i;

    cout << "s1 = " << s1 << ", s2 = "
         << s2 << ", s3 = " << s3 << endl;

    i = s2.length();
    if (s2.empty())
        cout << s2 << s1[i] << endl;
    else
        cout << s1.at(i + 1) << s2 << endl;

    if (s3.empty())
        s3 = s2 + s1.substr(5);
    else
        s3 = s2[2] + s1.substr(2, 3);

    s2 += "ed"

    cout << s1.substr(0, 5) << " " << s2 << " " << s3 << endl;

    return 0;
}
```

5. (20 points) Algorithmic complexity

For each function in this problem, determine (a) an equation for the worst-case computing time  $T(n)$  (expressed as a function of  $n$ , *i.e.*  $2n + 4$ ) and (b) the order of magnitude (expressed using big O notation, *i.e.*  $O(n)$ ). Note that:

- Each executable line of code is numbered so you can refer to it by number if necessary.
- A for loop may be treated as a single statement, not three separate statements.

a. (10 points)

```
int functionA(int n){
    int i;
1   int total = 1;
2   if (n > 0) {
3       for (i = n; i > 1; i--)
4           total = total * i;
    }
5   if (n < 0) {
6       for (i = n; i < 0; i++)
7           total = total + i;
    }
8   if (n % 2 == 0) {
9       for (i = 0; i < n; i += 2)
10          total = total + i;
    }
11  return total;
}
```

5 (continued)

For each function in this problem, determine (a) an equation for the worst-case computing time  $T(n)$  (expressed as a function of  $n$ , *i.e.*  $2n + 4$ ) and (b) the order of magnitude (expressed using big O notation, *i.e.*  $O(n)$ ). Note that:

- Each executable line of code is numbered so you can refer to it by number if necessary.
- A for loop may be treated as a single statement, not three separate statements.

b. (10 points)

```
void functionB(int arr[], int n){
    int i, j;
1   for (i = 0; i < n; i++) {
2       for (j = n - 1; j > 0; j = j / 2) {
3           if (arr[i] > arr[j])
4               arr[j] = arr[i];
           else
5               arr[i] = arr[j];
        }
    }
}
```

6. (16 points, 4 points each) Classes

For each of the multiple choice questions below, clearly indicate your response(s) by circling or underlining all choices you think best answer the question.

**NOTE: All parts of this question refer to the class E1Class, which is defined on the extra sheet included with the exam.**

a. Which of the following statements is a valid declaration for an object of type E1Class?

**This question has at least one correct answer, but may have more than one correct answer! Circle ALL choices that correctly answer the question.**

i. `E1Class ec1;`

ii. `E1Class ec2(1, 'a', 2.3);`

iii. `E1Class ec3(4, 5, 'b', 'c', 6.7, 8.9);`

iv. `E1Class ec4.changeAll(0, 0, 'x', 'y', 32.20, 2.25);`

v. `E1Class ec5.E1Class();`

b. Which of the following statements will compile and execute without errors, assuming the object `E1Class ec` has been properly initialized? **This question has at least one correct answer, but may have more than one correct answer! Circle ALL choices that correctly answer the question.**

i. `ec.changeAll(9, 8, '7', '6', 5.4, 3.2);`

ii. `ec.m1 = ec.m2;`

iii. `ec.printVars(cout);`

iv. `ec.E1Class(1, '2', 3.4);`

v. `int x = ec.someHelp(5);`

6 (continued)

c. Which of the following choices represents a valid implementation of one of the E1Class constructors? **This question has at least one correct answer, but may have more than one correct answer! Circle ALL choices that correctly answer the question.**

i. 

```
E1Class() {  
    m1 = 0;  
    m2 = 0;  
    m3 = 'e';  
    m4 = '1';  
    m5 = 1.5;  
    m6 = 9.9;  
}
```

ii. 

```
E1Class::E1Class() {  
    m1 = 5;  
    m2 = 6;  
    m3 = 'x';  
    m4 = m3;  
    m5 = m1;  
    m6 = m2;  
}
```

iii. 

```
E1Class::E1Class(int i, char c, double d) {  
    m1 = i;  
    m2 = i + 2;  
    m3 = c;  
    m4 = c + m1;  
    m5 = d;  
    m6 = d / 2;  
}
```

iv. 

```
E1Class::E1Class(int i1, int i2, char c1, char c2,  
                 double d1, double d2) {  
    m1 = i1;  
    m2 = i2;  
    m3 = c1;  
    m4 = c2;  
    m5 = d1;  
    m6 = d2;  
}
```

6 (continued)

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."

7. (10 points) **EXTRA CREDIT**

Complete the function below, which finds the longest substring shared by its two string arguments, s1 and s2. For example:

- `maxSubMatch("test 1", "test2")` returns "test"
- `maxSubMatch("EECE.3220", "19320")` returns "32"
- `maxSubMatch("test", "Exam")` returns "" (empty string)

```
string maxSubMatch(string s1, string s2) {
    string t1, t2;          // Temporary substrings for comparison
    string longest;        // Longest matching substring

    unsigned i, j, k;      // Loop indices
                           // ADD OTHER VARIABLES AS NEEDED

    // Go through both strings, comparing all substrings of
    //   same length, and find the longest one


```

```
    return longest;
}
```