

EECE 230 Introduction to Programming using C++ Final Exam (two hours),

December 17, 2013

Data types: (15 points)

Select from the listed options the data type that can *best represents* the following quantities.

- Number of citizens in a country.
a. `int n;` b. `unsigned int n;` c. `double n;` d. `int n[MAX];`
- Set of citizens in a country where `Citizen` is a C++ class.
a. `Citizen S;` c. `Citizen & S;`
b. `Citizen * S;` d. `Citizen S[MAX]; unsigned int size;`
- The area of a circle with radius r .
a. `int a;` b. `unsigned int a;` c. `double a;` d. `unsigned double a;`
- The credit/debit of a customer in a bank rounded to the unit currency bill.
a. `int c;` b. `unsigned int c;` c. `double c;` d. `unsigned double c;`
- Which of the below functions actually change(s) the values of x and y when called.

```
void swap1(int a, int b) {  
    int temp = a;  
    a = b;  
    b = a; }  
|  
|  
|
```

```
void swap2(int& a, int& b) {  
    int temp = a;  
    a = b;  
    b = a; }  
|  
|  
|
```

```
void swap3(int* a, int* b) {  
    int temp = *a;  
    *a = *b;  
    *b = temp;}  
|  
|  
|
```

- `swap1(x,y)`
- `swap2(x,y)`
- `swap3(&x,&y)`
- `swap2(x,y)` and `swap3(&x,&y)`
- `swap1(x,y)` and `swap2(x,y)`

Loops (18 points)

The code on the right requests a number between 0 and 127 inclusive from the user and stores the number in x .

Then the code makes several guesses asking the user each time whether the guess is correct, below x , or above x .

The code wins if it eventually guesses the number. The code loses if it terminates and fails to guess the number.

```
1 bool win = false;
2 int x, start=0, end=127, mid, answer;
3 cout << "Enter a number between 0 and 127 inclusive please: ";
4 cin >> x;
5 if (x < 0 || x > 127) {
6     cerr << "error, number out of range" << endl;
7     exit(-1); }
8 while (start <= end) {
9     mid = (start + end)/2;
10    cout << "My guess is " << mid << ", \n\t";
11    cout << "Please enter 0 if my guess is correct,\n\t"
12           "1 if my guess is bigger than your number,\n\t"
13           "and 2 if my guess is smaller than your number."<< endl;
14    cin >> answer;
15    if (answer == 0) { win = true; cout << "win"; break;}
16    else if (answer == 1) end = mid - 1;
17    else if (answer == 2) { start = mid+1;}
18    else { cerr << "answer has an illegal value." << endl; exit(-1); } }
19 if (win == false) {
20     cout << "program lost!" << endl;
21 } else {
22     cout << "program won!" << endl; }
```

6. Mark the correct statement given that the user provides legal input and does not cheat.

- a. The code always wins.
- b. The code runs forever.
- c. The code sometimes does not win.
- d. The code has compilation errors.

7. If the code runs, then the `while` loop makes at most k iterations before it terminates; where k is:

- a. 127
- b. 7
- c. $(0+127)/2$
- d. $(0+x+127)/2$
- e. ∞ (infinite loop)

8. If the user cheats, then the program can easily detect that on Line:

- a. The program can never detect cheating.
- b. Line 18
- c. Line 20
- d. Line 14

9. The following condition is equivalent to the condition on Line 5.

- a. $x \geq 0 \ \&\& \ x \leq 127$
- b. $!(x \geq 0 \ \&\& \ x \leq 127)$
- c. $0 \leq x \ \&\& \ x \leq 127$
- d. $!(0 \leq x \ \&\& \ x \leq 127)$
- e. options b and d.

10. The following Line is in clear violation of a programming style rule.

- a. Line 8
- b. Line 9
- c. Line 15
- d. Line 16
- e. None of the above

11. Assume $x \geq 0$. Which statement must be true after the following code is executed.

```
int y = 0; while (y < x) {
    y = y + 7;}
```

- a. y must be greater than 0
- b. y must be greater than x
- c. y may equal $x + 7$
- d. y may equal x
- e. y may equal 0

Arrays (5 points)

12. The best type to represent a valid index in an array is:
a. int **b.** unsigned int **c.** char **d.** unsigned char
13. When and how can we create an array of references to objects?
a. Never **d.** when objects are pointers: `int*objs;int*& refs[10]=*objs;`
b. Always **e.** when objects are dynamically allocated `int*& refs[10]=*new int[10];`
c. Only in main
14. When can a function return a value of type array?
a. In a class **b.** In a recursive function **c.** Never **d.** Always

The following on the right takes an array of integers a and its size n

```
1 int i=1, min=0;
2 while ( i <= n ) {
3     if(a[min] >= a[i])
4         {min = i};
5     i=i+1;}
```

15. When the loop terminates
a. $i = n$, and min is the minimum **d.** $i=n+1$, and $a[\text{min}]$ is the minimum
b. $i=n+1$, and min is the minimum **e.** the code fails if $a[0]$ is the minimum.
c. $i=n$, and $a[\text{min}]$ is the minimum
16. To make the code above find the maximum, we
a. rename variable min with max **d.** start i from $n - 1$ and decrement it
b. change line 3 to `if (a[i] >= a[min])` **e.** ! The code is buggy and needs rewriting!
c. start i at 0 instead of 1

More arrays (16 points)

The following memory diagram shows the state of a program at a point of its execution.

name:	a	n	a[0]																		
value:	α	12	4	4	5	5	6	6	6	7	7	7	8	8							
address:			α																		

```
int count = 0, int i = 1; while ( i < n ) {
    if (a[i] == a[i-1]) { count = count + 1;}
    i = i + 1;}
```

17. When the loop terminates `count` will have the value:
a. 4 **b.** 5 **c.** 6 **d.** 7
18. To count the unique numbers in the sorted array, change the `if` condition to:
a. `a[i] == a[i+1]` **d.** `a[i] < a[i-1]`
b. `a[i] != a[i+1]` **e.** can not be done by a simple change
c. `a[i] != a[i-1]`
19. The array *data* contains n elements. The code below is intended to shift the elements in *data* to the left with a wrap around. That is to convert

7	3	8	1	0	5
---	---	---	---	---	---

 into

3	8	1	0	5	7
---	---	---	---	---	---

```
statement 1
int i= 0; while ( i < n - 2 ) {
    data[i] = data[i+1]; i = i+1;}
statement 2
```

In order for the code to execute correctly statement 1 and statement 2 should be

- a. `temp = data[0]; ... data[0] = temp;`
- b. `temp = data[n-1]; ... data[0] = temp;`
- c. `temp = data[0]; ... data[n-1] = temp;` can not be done by a simple change
- d. `temp = data[n-1]; ... data[n-1] = temp;`

20. The function Z takes an array of characters `str` and its length `n` as arguments and returns a Boolean value.

```
bool Z(char str[], int n) {
    bool res = true;
    int i=0, j=n-1; while (i < j) {
        if (str[i] != str[j] ) { res = false; }
        i=i+1;
        j=j-1;
    }
    return res; }
```

We called function Z as follows `bool noon = Z(noon,4); bool madam = Z(madam,5);`.

- a. After the code is executed `noon` is *true* and `madam` is *false*.
- b. After the code is executed `noon` and `madam` are both `True`.
- c. After the code is executed `noon` is `False` and `madam` is `True`.
- d. After the code is executed `noon` and `madam` are both `False`.

Libraries (12 points)

21. Which of the following statements is incorrect about `rand`?
- a. `rand` returns a value between 0 and `RAND_MAX`
 - b. `rand` can be used to initialize large data sets for testing
 - c. `rand` can be used to approximate quantities such as areas under curves
 - d. A sequence of `n` calls to `rand` is guaranteed to always return a different sequence of numbers
22. The sequence of events to open a file using `ifstream` should be:
- a. declare the stream; call `open` (name of file); `read` ; close the file;
 - b. declare the stream; call `open` (name of file); check for errors; close the file;
 - c. declare the stream; call `open` (name of file); check for errors; `read`; close the file;
 - d. declare the stream; check for errors; `read`; close the file;
23. The `cin`, `cout`, and `cerr` are
- a. Reserved C++ keywords designating input, output, and error
 - b. global `iostream` objects linked to the console input/output devices
 - c. the compiler, the output, and the errors of visual studio
 - d. data types that can be used to declare several keyboards and screens
24. The `string` objects are dynamic arrays of characters that are
- a. equally replaceable by `vector<char>` where `vector` is a template dynamic array
 - b. highly compatible with the `iostream` and `fstream` libraries
 - c. always kept in order and sorted to speed up search
 - d. primitive objects of the C++ language

Simple code (15 points)

25. Write code that takes an array a of n elements and finds the range of a ; that is the difference between the largest element and the smallest element of a . For example, the range of the following array is $8 = 10 - 2$.

9	3	10	4	2	5
---	---	----	---	---	---

26. Given two values m and n , write code to find how many even numbers and how many odd numbers exist between m and n inclusive. For example, between 7 and 14 there are 4 odd numbers (7, 9, 11, 13), and (8, 10, 12, 14) and 4 even numbers.

Traffic control (25 points)

27. An intersection has (1) a name, (2) four traffic lights for cars, and (3) four traffic lights for pedestrians who want to cross the street. A car light has three colors designating its states: red, green, and yellow. The pedestrian light has two states: stop, and cross. Design three C++ classes, one for pedestrian lights, one for car lights, and one for intersections. The intersection class has a constructor that takes a name, starts all its car lights on red, and all its pedestrian lights on cross. The intersection class has a `start` method that runs forever. The `start` method simulates normal traffic controls where the lights change their states based on a timer control.

- I. Show a design of the classes
- II. Show an implementation of the constructor of the intersection class
- III. Show an implementation of the `start` method
- IV. Extend your work to have an array of intersections, where every three consecutive intersections synchronize and have their east/west car lights turn green simultaneously.

use this page for answers if needed.

Best of luck.