Fall Semester, Dr. Punch. Exam #2 (11/07), form 2 A

Last name (printed): ______________________________________________________________

First name (printed): ______________________________________________________________

Directions:

a) DO NOT OPEN YOUR EXAM BOOKLET UNTIL YOU HAVE BEEN TOLD TO BEGIN.
b) You have 90 minutes to complete the exam (7:00pm – 8:30pm)
c) This exam booklet contains 30 multiple choice questions, each weighted equally (5 points). **5, double-sided pages total**
d) You may use one 8.5” x 11” note sheet during the exam. No other reference materials or calculating devices may be used during the examination.
e) Questions will not be interpreted during the examination.
f) You should choose the single best alternative for each question, even if you believe that a question is ambiguous or contains a typographic error.
g) Please fill in the requested information at the top of this exam booklet.
h) Use a #2 pencil to encode any information on the OMR form.
i) Please encode the following on the OMR form:
   - Last name and first initial
   - MSU PID
   - Exam form (see the title of this page)
j) Please sign the OMR form.
k) Only answers recorded on your OMR form will be counted for credit.
l) Completely erase any responses on the OMR form that you wish to delete.
m) You must turn in this exam booklet and the OMR form when you have completed the exam. When leaving, please be courteous to those still taking the exam.

Good luck.

**Timing tip.** A rate of 2.5 minutes per multiple choice problem leaves 5 minutes to go over any parts of the exam you might have skipped.
```cpp
#include <iostream>
using std::cout; using std::endl;
#include <vector>
#include <string>

vector<string> fn1(vector<string> v1, vector<string> v2){
    vector<string> ans;
    for (size_t i = 0; i<v1.size(); ++i){
        ans.push_back(v1[i]);
        ans.push_back(v2[i]);
    }
    return ans;
}

string fn2(const vector<string>& v){
    string ans;
    for (auto i : v){                   // Line 1
        ans = i[0] + ans;
    }
    return ans;
}

void fn3(vector<string>& v, char c){
    for (auto i = v.begin(); i != v.end(); ++i){     // Line 2
        *i = *i + c;
    }
}

int main() {
    vector<string> v1 = {"abc", "def", "ghi"};
    vector<string> v2 = {"123", "456", "789"};
    auto fn1_ans = fn1(v1, v2);
    cout << fn1_ans.size() << endl;                    // Line 3

    auto fn2_ans = fn2(v1);
    cout << fn2_ans << endl;                           // Line 4

    fn3(v1, 'a');
    cout << v1.front() << endl;                       // Line 5
    cout << v1[0].back() << endl;                     // Line 6
}
```

Figure 1
1) What type is \( i \) on Line 1 of Figure 1?
   a) vector<string>
   b) string
   c) char
   d) size_t
   e) None of the above.

2) What type is \( i \) on Line 2 of Figure 1?
   a) vector<string>
   b) string
   c) char
   d) size_t
   e) None of the above.

3) What output is produced by Line 3 in Figure 1?
   a) 3
   b) 4
   c) 6
   d) 0
   e) None of the above.

4) What output is produced by Line 4 in Figure 1?
   a) abc
   b) adg
   c) gad
   d) empty string
   e) None of the above.

5) What output is produced by Line 5 in Figure 1?
   a) aabc
   b) aghi
   c) abc
   d) ghi
   e) None of the above.

6) What output is produced by Line 6 in Figure 1?
   a) a
   b) abc
   c) g
   d) ghi
   e) None of the above.
7) Which of the following are true about `default` designation on a constructor?
   a) Over-rides all user-provided constructors to use only the synthesized constructor.
   b) Sets the permissions of private data members to public
   c) For the default constructor, use the synthesized constructor
   d) All of the above
   e) None of the above

8) Which of the following are true about the * character in a C++ program?
   a) in the expression `1 * 2` it represents multiplication
   b) in the declaration `long* p` it represents dereferencing of a pointer
   c) in `cout << *p`, the * represents the creation of a pointer variable p.
   d) All of the above
   e) None of the above

9) Which of the following are true about a lambda function?
   a) it is a nameless function.
   b) they are often used in conjunction with STL algorithms
   c) they have a capture list
   d) All of the above
   e) None of the above

10) Which of the following are true about C++ constructors?
    a) They have the same name as the struct in which they are contained.
    b) They return the newly made element of the struct type
    c) They cannot take any arguments
    d) All of the above
    e) None of the above

11) Which of the following are true about a function which is templated?
    a) It is itself not a function, but a way to create a function.
    b) It contains the keyword `template`
    c) It makes use of a template parameter to represent a calling type.
    d) All of the above
    e) None of the above

12) Which of the following is true about a function parameter which is both a const and a reference?
    a) A copy is made of that parameter when the function is invoked.
    b) You cannot change the parameter inside the function
    c) It only works with pointer types.
    d) All of the above
    e) None of the above
// for brevity, let's assume I got the includes correct

string fn1(const vector<string>& v, string s){
    auto pos = find(v.begin(), v.end(), s);
    return (pos == v.end()) ? "nope" : *pos;
}

size_t fn2(vector<string>& v, string s){
    string ans;
    auto pos = find(v.begin(), v.end(), s);              // Line 1
    if (pos != v.end())
        pos = find(pos+1, v.end(), s);
    return pos - v.begin();
}

string trans_fn(string s){
    string ans;
    for (auto e : s)
        ans += toupper(e);
    return ans;
}

vector<string> fn3(const vector<string>& v){
    vector<string> ans;
    transform(v.begin(), v.end(), back_inserter(ans), trans_fn);
    return ans;
}

bool sort_fn(const string& s1, const string& s2){
    return s1.back() < s2.back();
}

void fn4(vector<string>& v){
    sort(v.begin(), v.end(), sort_fn);
}

int main(){
    vector<string> v{"bill", "fred", "bill", "george" };
    cout << fn1(v, "Bill") << endl;        // Line 2
    cout << fn2(v, "bill") << endl;        // Line 3
    auto ans = fn3(v);                     // Line 4
    cout << ans[0][0] << endl;             // Line 5
    fn4(v);
    cout << v.front() << endl;             // Line 6
}

Figure 2
13) For the program in Figure 2, what type is pos on Line 1.
   a) size_t
   b) char
   c) vector<string>::iterator
   d) vector<string>
   e) None of the above

14) What output is produced by Line 2 in Figure 2?
   a) nope
   b) bill
   c) bill, fred, bill, george
   d) Bill
   e) None of the above

15) What output is produced by Line 3 of Figure 2?
   a) 3
   b) 2
   c) 1
   d) 0
   e) None of the above

16) For the program in Figure 2, what type is ans on Line 4?
   a) size_t
   b) char
   c) vector<string>::iterator
   d) vector<string>
   e) None of the above

17) What output is produced by Line 5 in Figure 2?
   a) b
   b) B
   c) bill
   d) Bill
   e) None of the above

18) What output is produced by Line 6 in Figure 2?
   a) bill
   b) fred
   c) george
   d) empty string
   e) None of the above
Figure 3

19) What output is produced by Line 1 in Figure 3?
   a) a  
   b) b  
   c) c  
   d) d  
   e) None of the above
20) What output is produced by Line 2 in Figure 3?
   a) a
   b) aab
   c) a:aab
   d) aabc
   e) None of the above
21) What output is produced by Line 3 of Figure 3?
   a) 0
   b) 1
   c) 2
   d) 3
   e) None of the above
22) What output is produced by Line 4 of Figure 3?
   a) 0
   b) 1
   c) 2
   d) 3
   e) None of the above
23) What output is produced by Line 5 of Figure 3?
   a) 0
   b) 1
   c) 2
   d) 3
   e) None of the above
24) What output is produced by Line 6 of Figure 3?
   a) bbc
   b) bbcc
   c) bbccw
   d) b
   e) None of the above
```cpp
// let's assume I got the includes right
struct MyStruct{
    long lng = 1;
    string str = "1";

    MyStruct() = default;
    MyStruct(long l, string s) : lng(l), str(s) {};

    string method1();
    long method2(string);
    MyStruct method3(const MyStruct&);
};

string MyStruct::method1()
    return to_string(lng) + ":" + str;

long MyStruct::method2(string s){
    str = str + s;
    for (auto c : s)
        lng += c - '0'; // helpful comment, mind the 
    return s.size();
}

MyStruct MyStruct::method3(const MyStruct& m){
    MyStruct ans(lng, str);
    ans.method2(m.str);
    return ans;
}

int main(){
    MyStruct ms1(6, "123");
    cout << ms1.method1() << endl; // Line 1
    MyStruct ms2;
    cout << ms2.method1() << endl; // Line 2

    cout << ms1.method2("345") << endl; // Line 3
    cout << ms1.method1() << endl; // Line 4

    MyStruct ms3(5, "14");
    MyStruct ms4(3, "12");
    auto temp = ms3.method3(ms4);
    cout << temp.method1() << endl; // Line 5
    cout << ms4.method1() << endl; // Line 6
}
```

Figure 4
25) For the program in Figure 4, what value is printed by Line 1?
   a) 6
   b) 123
   c) 6:123
   d) 123:6
   e) None of the above

26) For the program in Figure 4, what value is printed by Line 2?
   a) 1:1
   b) 0:0
   c) 0
   d) 1
   e) None of the above

27) For the program in Figure 4, what value is printed by Line 3?
   a) 0
   b) 1
   c) 2
   d) 3
   e) None of the above

28) For the program in Figure 4, what value is printed by Line 4?
   a) 6:123
   b) 12:345
   c) 7:1231
   d) 18:123345
   e) None of the above

29) For the program in Figure 4, what value is printed by Line 5?
   a) 5:14
   b) 3:12
   c) 8:1412
   d) 1:1
   e) None of the above

30) For the program in Figure 4, what value is printed by Line 6?
   a) 5:14
   b) 3:12
   c) 8:1412
   d) 1:1
   e) None of the above