Spring Semester, Dr. Punch. Exam #2 (03/28), form 2 B

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.
#include<string>
using std::string;
#include<iostream>
using std::endl; using std::cout; using std::ostream;

struct MyStruct{
    public:
        long a = 0;
        long b = 0;
        long c = 0;

        MyStruct() = default;
        MyStruct(long p1, long p2, long p3) : a(p1), b(p2), c(p3) {};

        long method1 (long);
        MyStruct method2 (MyStruct);
        ostream& method3(ostream&);
};

long MyStruct::method1(long x){
    return (a*x*x + b*x + c);
}

MyStruct MyStruct::method2(MyStruct ms){
    return MyStruct(a + ms.a, b + ms.b, c + ms.b);
}

ostream& MyStruct::method3(ostream& out){
    out << a << "x**2 + " << b << "x + ""<<c;
    return out;
}

int main (){
    MyStruct ms0;
    cout << ms0.a << endl;   // Line 1
    MyStruct ms1(1,2,3);
    cout << ms1.b << endl;   // Line 2
    cout << ms1.method1(2) << endl; // Line 3
    auto result1 = ms1.method2(ms0);
    cout << result1.c << endl;  // Line 4
    ms1.method3(cout) << endl;  // Line 5
}

Figure 4

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

3) For the program in Figure 4, what value is printed by Line 3?
   a) 11
   b) 10
   c) 6
   d) 0
   e) None of the above

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

5) For the program in Figure 4, what value is printed by Line 5?
   a) empty string
   b) 0x**2 + 0x + x
   c) 1x**2 + 2x + 3
   d) 3x**2 + 2x + 1
   e) None of the above


```cpp
#include <iostream>
using std::cout; using std::endl;
#include <vector>
using std::vector;
#include <iterator>
using std::back_inserter;
#include <algorithm>
using std::copy;

long fn1(vector<long>& v){
  long result = v.front();
  for (size_t i = 1; i < v.size() - 1; ++i)
    v[i-1] = v[i];
  v.pop_back();
  return result;
}

size_t fn2(vector<long>& v1, vector<long>& v2){
  size_t result = v2.size();
  copy(v2.begin(), v2.end(), back_inserter(v1));
  return result;
}

int main(){
  vector<long> v1 = {2,4,6,8};
  cout << fn1(v1) << endl;   // Line 1
  cout << v1.size() << endl; // Line 2
  cout << v1.front() << endl; // Line 3

  vector<long> v2 = {1,3,5,7,9};
  vector<long> v3 = {2,4,6};

  auto result = fn2(v2,v3); // Line 4
  cout << result << endl;    // Line 5
  cout << v2.size() << endl; // Line 6
  cout << v2.back() << endl;  // Line 7
}
```

Figure 1

6) What output is produced by Line 1 in Figure 1?
   a) 1  
   b) 2  
   c) 3  
   d) 4  
   e) None of the above.
7) What output is produced by Line 2 in Figure 1?
   a) 1
   b) 2
   c) 3
   d) 4
   e) None of the above.

8) What output is produced by Line 3 in Figure 1?
   a) 1
   b) 2
   c) 3
   d) 4
   e) None of the above.

9) What type is result on Line 4 in Figure 1?
   a) vector<long>
   b) long
   c) int
   d) string
   e) None of the above.

10) What output is produced by Line 5 in Figure 1?
    a) 1
    b) 2
    c) 3
    d) 4
    e) None of the above.

11) What output is produced by Line 6 in Figure 1?
    a) 3
    b) 4
    c) 5
    d) 8
    e) None of the above.

12) What output is produced by Line 7 in Figure 1?
    a) 6
    b) 5
    c) 4
    d) 8
    e) None of the above.
13) Which of the following clears the input buffer \texttt{cin} of all remaining input?
   a) \texttt{cin.clear}
   b) \texttt{cin.ignore}
   c) \texttt{cin.reset}
   d) All of the above
   e) None of the above

14) Which of the following is true about an \texttt{ostringstream} variable \texttt{sout}?
   a) you can output to \texttt{sout} using \texttt{<<} operations.
   b) it is a string so you can concatenate to it, for example \texttt{sout + "my string".}
   c) you \textbf{cannot} use format manipulators, for example
      \texttt{sout \textless\textless setprecision(4) \textgreater\textgreater \text{is not allowed}}
   d) All of the above.
   e) None of the above.

15) What is meaning of the keyword \texttt{auto} in a type declaration?
   a) Can determine a variable's type at run time.
   b) Does not actually set a variable's type. It is a way around the type system.
   c) Is restricted to only built-in types like \texttt{long} or \texttt{string}.
   d) All of the above
   e) None of the above

16) Which of the following generic algorithms allows you to modify their "standard operation" by adding a function of your own design?
   a) \texttt{sort}
   b) \texttt{accumulate}
   c) \texttt{transform}
   d) All of the above
   e) None of the above

17) What does \texttt{=delete} mean at the end of a constructor declaration?
   a) that constructor is not allowed
   b) that constructor is deleted in the header but is defined elsewhere
   c) that constructor cannot take any arguments (it is a default constructor)
   d) that constructor is a \texttt{delete} constructor, it deletes elements from the class
   e) None of the above
#include <iostream>
using std::cout; using std::endl;
#include <map>
using std::map;
#include <string>
using std::string;

long fn1 (map<string, long>& m) {
    long result = 0;
    for (auto e : m) {       // Line 1
        result += e.second;
    }
    return result;
}

long fn2 (map<string, long>& m, string str, long lng) {
    auto i = m.find (str);   // Line 2
    if (i != m.end())
        i->second += lng;
    else
        m[str] = lng;
    return m[str];
}

int main () {
    map<string, long> m { "bill", 1 }, { "fred", 2 }, { "jane", 3 };
    cout << fn1 (m) << endl;   // Line 3
    cout << fn2 (m, "ginger", 0) << endl; // Line 4
    cout << m.size() << endl; // Line 5
    cout << fn2 (m, "bill", 5) << endl; // Line 6
}

18) For the program in Figure 2, what type is e on Line 1.
   a) char
   b) map<string, long>::iterator
   c) map<string, long>*
   d) string
   e) None of the above
19) For the program in Figure 2, what type is i on Line 2.
   a) char
   b) map<string, long>::iterator
   c) map<string, long>*
   d) string
   e) None of the above
20) What output is produced by Line 3 of Figure 2?
   a) 3  
   b) 4  
   c) 5  
   d) 6  
   e) None of the above

21) What output is produced by Line 4 of Figure 2?
   a) 3  
   b) 4  
   c) 5  
   d) 6  
   e) None of the above

22) What output is produced by Line 5 of Figure 2?
   a) 3  
   b) 4  
   c) 5  
   d) 6  
   e) None of the above

23) What output is produced by Line 6 in Figure 2?
   a) 3  
   b) 4  
   c) 5  
   d) 6  
   e) None of the above
24) For the program in Figure 3, what type is \textit{i} on Line 1.
   a) vector<long>
   b) string
   c) long
   d) char
   e) None of the above
25) What output is produced by Line 2 in Figure 3?
   a) true
   b) false
   c) 0
   d) 1
   e) None of the above

26) What output is produced by Line 3 of Figure 3?
   a) 2
   b) 3
   c) 4
   d) 5
   e) None of the above

27) What output is produced by Line 4 of Figure 3?
   a) 2
   b) 3
   c) 4
   d) 5
   e) None of the above

28) What output is produced by Line 5 of Figure 3?
   a) 2
   b) 3
   c) 4
   d) 5
   e) None of the above

29) What output is produced by Line 6 of Figure 3?
   a) 2
   b) 3
   c) 4
   d) 5
   e) None of the above

30) What output is produced by Line 7 of Figure 3?
   a) empty string
   b) 1, 2, 3, 4, 5,
   c) 1:2:3:4:5
   d) 1:2:3:4:
   e) None of the above