Fall Semester, Dr. Punch. Exam #2 (11/09), form 2 A

Last name (printed):____

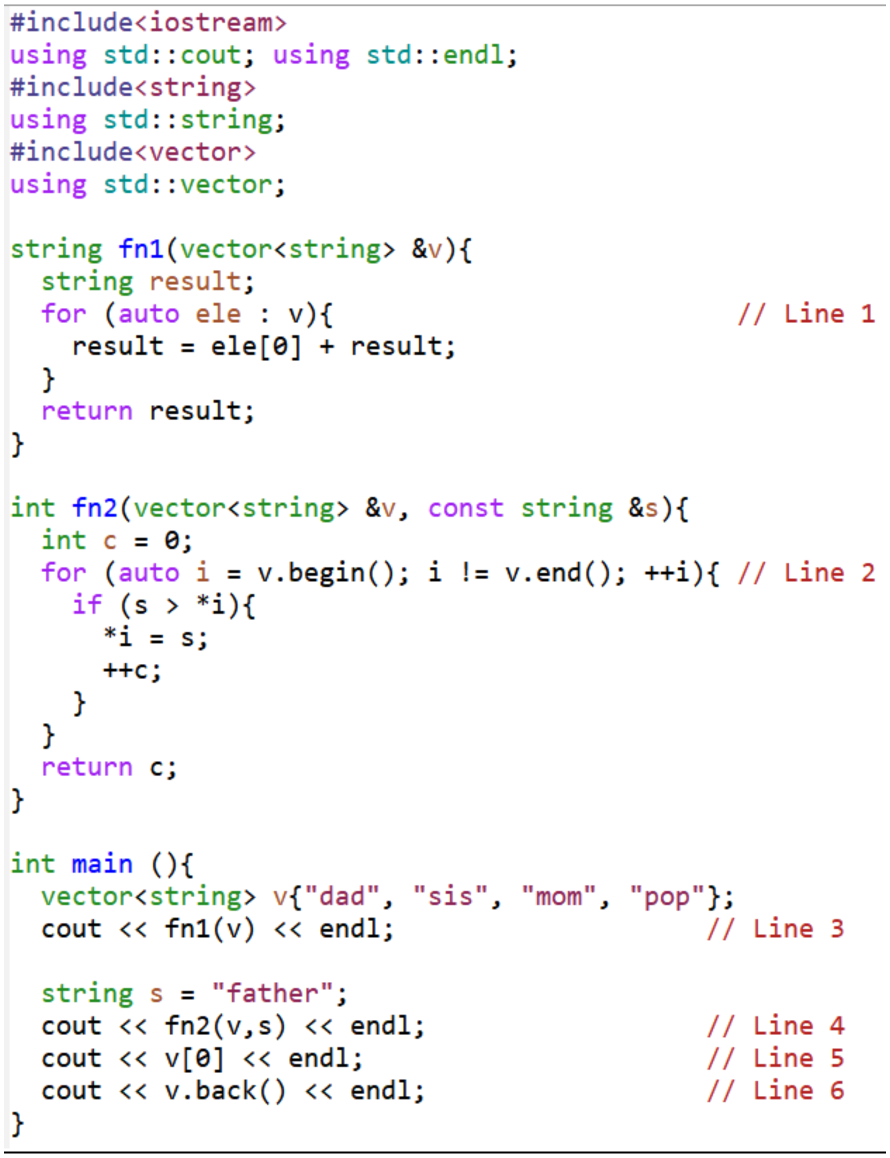
First name (printed):____

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). **<u>5, double-sided pages total</u>**
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.

<u>Timing tip</u>. 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<iostream>
using std::cout; using std::endl;
#include<string>
using std::string;
#include<vector>
using std::vector;

string fn1(vector<string> &v){
  string result;
  for (auto ele : v){                         // Line 1
    result = ele[0] + result;
  }
  return result;
}

int fn2(vector<string> &v, const string &s){
  int c = 0;
  for (auto i = v.begin(); i != v.end(); ++i){ // Line 2
    if (s > *i){
      *i = s;
      ++c;
    }
  }
  return c;
}

int main (){
  vector<string> v{"dad", "sis", "mom", "pop"};
  cout << fn1(v) << endl;                    // Line 3

  string s = "father";
  cout << fn2(v,s) << endl;                  // Line 4
  cout << v[0] << endl;                      // Line 5
  cout << v.back() << endl;                  // Line 6
}
```

Figure 1

1) What type is `ele` on `Line 1` of Figure 1?
   a) `vector`
   b) `vector<string>`
   c) `vector<string>::iterator`
   d) `vector<string>*`
   e) None of the above.

E

2) What type is `i` on `Line 2` of Figure 1?
   a) `vector`
   b) `vector<string>`
   c) `vector<string>::iterator`
   d) `vector<string>*`
   e) None of the above.

C

3) What output is produced by `Line 3` in Figure 1?
   a) `dsmp`
   b) `pmsd`
   c) `dadsismompop`
   d) `dad`
   e) None of the above.

B

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

A

5) What output is produced by `Line 5` in Figure 1?
   a) `dad`
   b) `sis`
   c) `mom`
   d) `pop`
   e) None of the above.

E

6) What output is produced by `Line 6` in Figure 1?
   a) `dad`
   b) `sis`
   c) `mom`
   d) `pop`
   e) None of the above.

D

7) Which of the following are true about C++ functions?

C

a) You cannot have more than one function with the same name.
b) a function must have at least one parameter
c) C++ uses the types of the parameters and return value to differentiate functions with the same name
d) the first parameter of a function does not require any type information
e) None of the above

8) Which of the following are true about STL iterators?

A

a) you can treat them as if they were pointers
b) you cannot use them in conjunction with an STL container
c) iterators are not templated elements.
d) All of the above
e) None of the above

9) Which of the following are true about a `lambda` construct?

A

a) it is a function and can be invoked like a function
b) it has a name
c) it cannot be used in a generic algorithm
d) All of the above
e) None of the above

10) Which of the following are true about C++ constructors?

C

a) They cannot be defined inside of a `struct`.
b) They require a `return` statement to return newly made `struct` type
c) They can be overloaded for different parameter lists
d) All of the above
e) None of the above

11) Which of the following are correct about methods?

D

a) A method can be part of a `struct`
b) It is called in the context of an object using a dot call
c) In calling a method the `this` pointer is assigned by the compiler
d) All of the above
e) None of the above

12) Which of the following is true about a variable `m` of type `map<string,long>`?

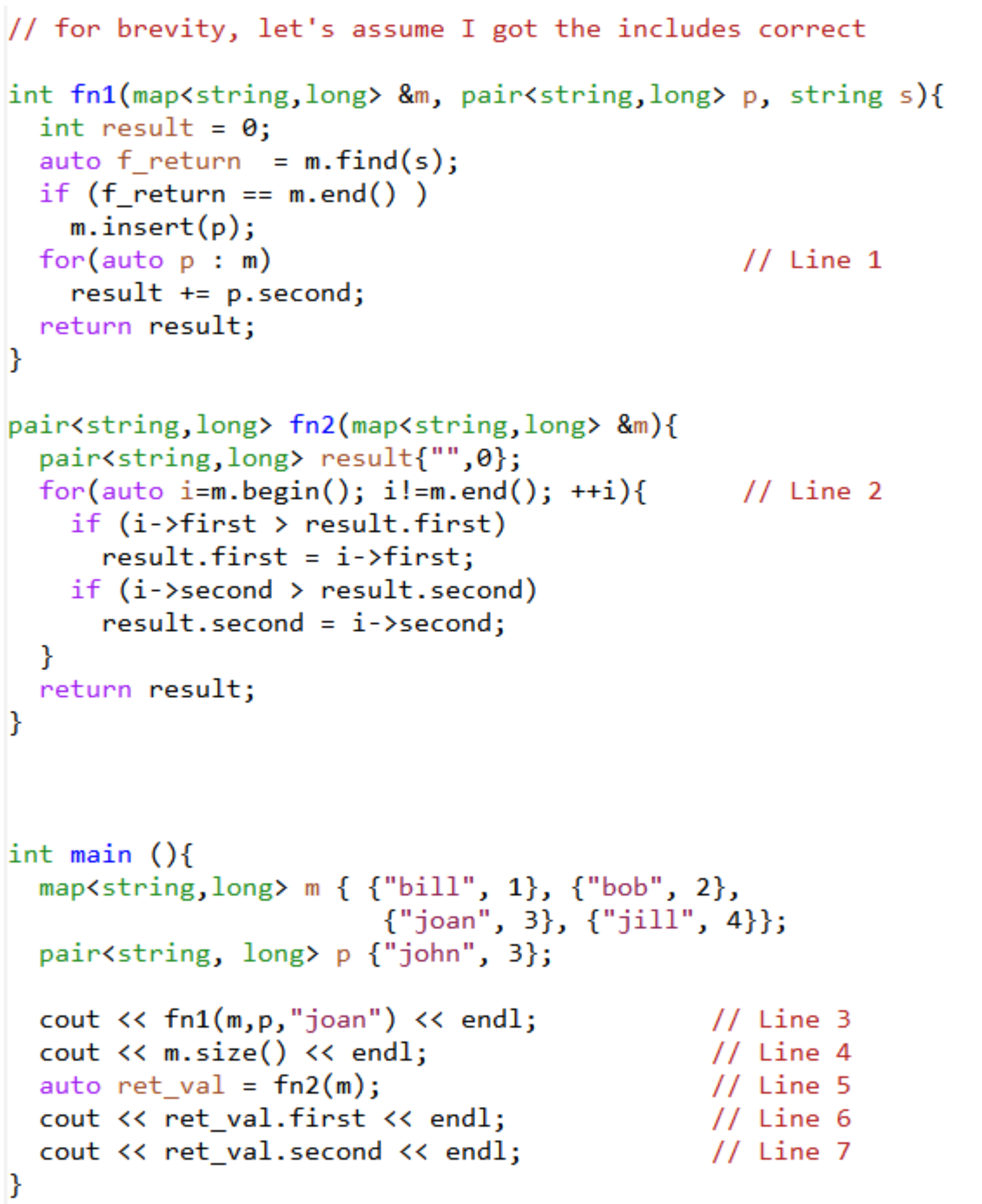
B

a) `m` has no order to its elements
b) `m["abc"] = 2` assigns the value 2 to the key `"abc"`
c) `cout << m` is a legal operation.
d) All of the above.
e) None of the above.

13) Which of the following generic algorithms would you use to multiply all the values of a `vector<long>` together into a single result

D

a) copy
b) transform
c) sort
d) accumulate 求和
e) None of the above



```
// for brevity, let's assume I got the includes correct

int fn1(map<string,long> &m, pair<string,long> p, string s){
  int result = 0;
  auto f_return  = m.find(s);
  if (f_return == m.end() )
    m.insert(p);
  for(auto p : m)                             // Line 1
    result += p.second;
  return result;
}

pair<string,long> fn2(map<string,long> &m){
  pair<string,long> result{"",0};
  for(auto i=m.begin(); i!=m.end(); ++i){     // Line 2
    if (i->first > result.first)
      result.first = i->first;
    if (i->second > result.second)
      result.second = i->second;
  }
  return result;
}



int main (){
  map<string,long> m { {"bill", 1}, {"bob", 2},
                       {"joan", 3}, {"jill", 4}};
  pair<string, long> p {"john", 3};

  cout << fn1(m,p,"joan") << endl;          // Line 3
  cout << m.size() << endl;                 // Line 4
  auto ret_val = fn2(m);                    // Line 5
  cout << ret_val.first << endl;            // Line 6
  cout << ret_val.second << endl;           // Line 7
}
```

Figure 2

14) For the program in Figure 2, what type is `p` on `Line 1`.

a) `string`

b) `map<string,long>::iterator`

c) `map<string,long>*`

d) `pair<string,long>::iterator`

e) None of the above

E

pair<string, long>

15) What type is `i` on `Line 2` in Figure 2?

B

a) `string`
b) `map<string,long>::iterator`
c) `map<string,long>*`
d) `pair<string,long>::iterator`
e) None of the above

16) What output is produced by `Line 3` in Figure 2?

E

a) `1`
b) `2`
c) `3`
d) `4`
e) None of the above

17) What output is produced by `Line 4` of Figure 2?

A

a) `4`
b) `3`
c) `2`
d) `1`
e) None of the above

18) What type is `ret_val` on `Line 5` of Figure 2?

C

a) `long`
b) `string`
c) `pair<string,long>`
d) `map<string,long>::iterator`
e) None of the above

19) What output is produced by `Line 6` in Figure 2?

C

a) `bill`
b) `bob`
c) `joan`
d) `jill`
e) None of the above

20) What output is produced by `Line 7` in Figure 2?

D

a) `1`
b) `2`
c) `3`
d) `4`
e) None of the above

```
// for brevity, let's assume I got the includes right
string fn1(vector<string> &v){
  sort(v.begin(), v.end(),
          [] (const string &s1, const string &s2){
            return s1[1] < s2[1];
          }
      ); // of sort
    return v.front();
}

string fn2(const string &s){
  return s + ":" +  to_string(s.size() );
}

vector<string> fn3(vector<string> &v){
  vector<string> result;
  transform (v.begin(), v.end(), back_inserter(result), fn2 );
  return result;
}

bool fn4(string s){
  return s.size() > 5;
}

string fn5(const vector<string> &v){
  ostringstream oss;
  copy_if(v.cbegin(), v.cend(), ostream_iterator<string>(oss, ","), fn4);
  string result = oss.str();
  return result.substr(0, result.size()-1);
}

int main (){
  vector<string> v{"brahms", "bach", "listz", "copland"};
  cout << fn1(v) << endl;                // Line 1
  cout << v.back() << endl;              // Line 2

  v = {"brahms", "bach", "listz", "copland"};
  auto result = fn3(v);                  // Line 3
  cout << result[0] << endl;             // Line 4
  cout << fn5(v) << endl;                // Line 5
}
```

Sort 算法
匿名函数

升序

Figure 3

21) What output is produced by `Line` 1 of Figure 3?

B

a) `brahms`
b) `bach`
c) `listz`
d) `copland`
e) None of the above

22) What output is produced by `Line2` in Figure 3?

A

a) `brahms`
b) `bach`
c) `listz`
d) `copland`
e) None of the above

23) For the program in Figure 3, what type is `result` on `Line 3`?

A

a) `vector<string>`
b) `string`
c) `long`
d) `char`
e) None of the above

24) What output is produced by `Line 4` of Figure 3?

C

a) `:6`
b) `brahms`
c) `brahms:6`
d) `bach`
e) None of the above

25) What output is produced by `Line 5` of Figure 3?

B

a) `brahms`
b) `brahms,copland`
c) `brahms,listz,copland`
d) `brahms,bach,listz,copland`
e) None of the above

```
// assume correct includes
struct MyStruct{

  string s_member;
  long l_member;
  map<string,long> m_member;

  MyStruct() =default;
  MyStruct(map<string,long>);
  void method1(string, long);
  int method2();
  MyStruct method3(const MyStruct&);
};

MyStruct::MyStruct(map<string,long> m){
  m_member = m;
  s_member = m.begin() -> first;
  l_member = m.begin() -> second;
}

void MyStruct::method1(string s, long l){
  pair<string,long> p(s,l);
  m_member.insert(p);
}

int MyStruct::method2(){
  int result=0;
  for (auto ele : m_member){
    result += ele.second;
  }
  return result;
}

MyStruct MyStruct::method3(const MyStruct &m1){
  MyStruct result;
  for (auto ele : m1.m_member){
    if (m_member.find(ele.first) !=  m_member.end() ){
      result.m_member.insert(ele);
    }
  }
  result.s_member = result.m_member.begin() -> first;
  result.l_member = result.m_member.begin() -> second;
  return result;
}

int main (){
  MyStruct ms1( { {"bill",1}, {"alex",2}, {"fred",3} } );
  cout << ms1.s_member << endl;                // Line 1
  ms1.method1("fred", 4);
  cout << ms1.m_member.size() << endl;         // Line 2

  MyStruct ms2( { {"john",5}, {"bill", 6} } );
  cout << ms2.method2() << endl;               // Line 3
  auto result = ms1.method3(ms2);
  cout << result.m_member.size() << endl;      // Line 4
  cout << result.l_member << endl;             // Line 5
}
```

C++ Map
会自动排序

Figure 4

26) For the program in Figure 4, what value is printed by `Line 1`?

B ~~A~~

a) `bill`
b) `alex`
c) `fred`
d) `empty string`
e) None of the above

27) For the program in Figure 4, what value is printed by `Line 2`?

C

a) `1`
b) `2`
c) `3`
d) `6`
e) None of the above

28) For the program in Figure 4, what value is printed by `Line 3`?

E

a) `1`
b) `2`
c) `3`
d) `6`
e) None of the above

29) For the program in Figure 4, what value is printed by `Line 4`?

A

a) `1`
b) `2`
c) `3`
d) `6`
e) None of the above

30) For the program in Figure 4, what value is printed by `Line 5`?

D

a) `1`
b) `2`
c) `3`
d) `6`
e) None of the above