Fall Semester 16, Dr. Punch. Exam #2 (11/10), 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 80 minutes to complete the exam (10:20-11:40)
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>
using std::vector;
#include<algorithm>
using std::find;

vector<long> fn1(const vector<long>& v1, const vector<long>& v2){
  vector<long> result;
  for(auto e : v1){                              // Line 1
    auto i = find(v2.begin(), v2.end(), e);      // Line 2
    if (i != v2.end())
      result.push_back(e);
  }
  return result;
}

vector<long> fn2(const vector<long>& v1, const vector<long>& v2){
  size_t i=0, j=0;
  vector<long> result;
  while ( i < v1.size() && j < v2.size() ){
    if (v1[i] < v2[j]){
      result.push_back(v1[i]);
      ++i;
    }
    else{
      result.push_back(v2[j]);
      ++j;
    }
  }
  return result;
}

int main (){
  vector<long> v1{1,2,3};
  vector<long> v2{1,0,2,0};

  auto v = fn1(v1,v2);
  cout << v.size() << endl;       // Line 3
  cout << v.front() << endl;      // Line 4
  v = fn2(v1,v2);
  cout << v.size() << endl;       // Line 5
  cout << v.back() << endl;       // Line 6
}
```

Figure 1

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

D

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

C

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

C

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

D

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

E

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

E

7) Which of the following are true about a `friend` designation?
   a) It's a push, not a pull
   b) it indicates permission to access private class elements
   c) it can apply to both a class and a function.
   d) All of the above
   e) None of the above

8) What is the difference between capacity and size in a vector?
   a) You need to know the type of the vector to answer
   b) size is how many elements are in the vector, capacity is how many it **could** hold before growing
   c) capacity is how many elements are in the vector, size is how many it **could** hold before growing
   d) size is how many elements are in the vector, capacity is maximum it **could ever** hold.
   e) None of the above

9) Which of the following are true of the of `cin.ignore` method?
   a) removes characters from the input stream.
   b) ignores the last cin operation
   c) clears any errors, resets cin to be "good"
   d) closes the cin stream
   e) None of the above

10) Which of the following are true about the C++ `map` STL container?
   a) It's a sequence.
   b) It consists **only** of a series of keys.
   c) Requires a single template type to make a variable of type `map`.
   d) All of the above
   e) None of the above

11) Which of the following are true about the operator *?
   a) As a binary operation it represents multiply.
   b) In a declaration it represents a pointer type.
   c) As a unary operation it represents de-reference.
   d) All of the above
   e) None of the above

12) Which of the following are true about the special variable `this`?
   a) You can **directly** assign it a new value in a class's method.
   b) It is automatically set by C++ to the address of the calling variable.
   c) It is an integer type
   d) All of the above
   e) None of the above

13) Which of the following are true about C++ random numbers, as discussed in class?
   a) they generate the same sequence when starting from the same seed.
   b) when used in conjunction with a distribution, the distribution uses the random number generator as an argument.
   c) C++ has multiple versions of random number generators;
   d) All of the above
   e) None of the above

```
#include<iostream>
using std::cout; using std::endl;
#include<string>
using std::string;
#include<map>
using std::map;
#include<utility>
using std::pair;

long fn1(map<long,string>&m, string s){
  long res=0;
  for(auto e : s)                                    // Line 1
    res += static_cast<long>(e - '0');
  m[res]=s;
  return res;
}

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

int main (){
  map<long,string> m{ {2,"11"}, {3,"102"} };
  auto l = fn1(m,"7890");
  cout << l << endl;                                 // Line 3
  cout << m[l] << endl;                              // Line 4
  cout << m.size() << endl;                          // Line 5
  auto p = fn2(m, "8");
  cout << p.first << endl;                           // Line 6
  cout << p.second << endl;                          // Line 7
}
```

Figure 2

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

A

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

15) For the program in Figure 2, what type is `itr` in `Line 2`.

C

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

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

B

a) `4`
b) `24`
c) `7890`
d) `0`
e) None of the above

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

C

a) `4`
b) `24`
c) `7890`
d) `0`
e) None of the above

18) What output is produced by `Line 5` in Figure 2?

D

a) `0`
b) `1`
c) `2`
d) `3`
e) None of the above

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

E

a) `0`
b) `1`
c) `2`
d) `3`
e) None of the above

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

A

a) `0`
b) `1`
c) `2`
d) `3`
e) None of the above

```cpp
#include<iostream>
using std::cout; using std::endl; using std::ostream;
#include<string>
using std::string; using std::to_string;

class MyClass{
private:
  string s1_;
  long l_;

public:
  MyClass()=default;                    // Line 1
  MyClass(long l, string s);
  void do_it(long);
  friend MyClass operator+ (const MyClass&, const MyClass&);
  friend ostream& operator<<(ostream&, const MyClass&);
};

MyClass::MyClass(long l, string s){
  l_ = l;
  s1_ = s + to_string(l);
}

void MyClass::do_it(long param_l){
  s1_ += to_string(param_l);
  l_ += param_l;
}
```

```cpp
ostream& operator<<(ostream& o, const MyClass& c){
  o << c.s1_ << ":" << c.l_;
  return o;
}

MyClass operator+(const MyClass& c1, const MyClass& c2){
  string s;
  s = (c1.s1_ > c2.s1_) ? c1.s1_ + c2.s1_ : c2.s1_ + c1.s1_;
  return MyClass(c1.l_ + c2.l_, s);
}

int main(){
  MyClass mc1(3, "a");
  cout << mc1 << endl;                  // Line 2
  mc1.do_it(12);
  cout << mc1 << endl;                  // Line 3
  MyClass mc2(7, "c");
  MyClass mc3(5, "b");
  cout << mc2 + mc3 << endl;            // Line 4
  MyClass mc4(9, "d");
  cout << mc4 + mc2 + mc3 << endl;      // Line 5
}
```

Figure 3

21) For the program in Figure 3, which of the following are true about `=default` on `Line 1`?

A

已定义一个有参的构造函数
则默认构造函数编译器不再生成

a) Use the C++ default constructor
b) It means the constructor cannot be called implicitly by C++
c) The class can have no private values
d) All of the above
e) None of the above

22) For the program in Figure 3, give the output of `Line 2`?

C

a) `a3`
b) `3`
c) `a3:3`
d) `3:a3`
e) None of the above

23) For the program in Figure 3, give the output of `Line 3`?

A

a) `a312:15`
b) `a3:12`
c) `a3:15`
d) `12a3:15`
e) None of the above

24) For the program in Figure 3, give the output of `Line 4`?

a) `c7b5:12`
b) `c7b512:12`
c) `b5c7:12`
d) `c7b512:15`
e) None of the above

B

25) For the program in Figure 3, give the output of `Line 5`?

a) `d9c7b5:21`
b) `b5d9c7:21`
c) `d9c716b521:21`
d) `b5c7d9:21`
e) None of the above

C

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

struct AStruct{
  vector<long> v;
  long val;
};

void fn1(AStruct& a, vector<long> param_v){
  a.v = param_v;
  a.val = a.v.front();
}

void fn2(AStruct& a, long l){
  vector<long> new_v={l};
  a.val = l;
  copy(a.v.begin(), a.v.end(), back_inserter(new_v));
  a.v = new_v;
}

string fn3(AStruct& a){
  ostringstream oss;
  oss << a.v.front() << ":" << a.v.back();
  return oss.str();
}
```

```cpp
void fn4(AStruct& a, long l){
  long t;
  int i,j;
  size_t sz = a.v.size();
  for(i=0; i < l; i++){
    t = a.v.front();
    for(j=1; j < sz-1; j++){
      a.v[j-1] = a.v[j];
    } // of j loop
    a.v[sz-1] = t;
  } // of i loop
  a.val = a.v.front();
}

int main (){
  AStruct a;
  vector<long> v = {1,2,3};
  fn1(a,v);
  cout << a.val << endl;      // Line 1
  cout << fn3(a) << endl;     // Line 2
  fn2(a,5);
  cout << fn3(a) << endl;     // Line 3
  fn4(a,2);
  cout << a.val << endl;      // Line 4
  cout << fn3(a) << endl;     // Line 5
}
```

Figure 4

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

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

A

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

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

B

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

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

E

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

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

B

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

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

D