Directions:

a) DO NOT OPEN YOUR EXAM BOOKLET UNTIL YOU HAVE BEEN TOLD TO BEGIN.

b) You have 120 minutes to complete the exam (8pm-10pm)

c) This exam booklet contains 40 multiple choice questions, each weighted equally (5 points). **6 double-sided pages in 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 2.75 minutes per multiple choice problem leaves 10 minutes to go over any parts of the exam you might have skipped.
#include <iostream>
using std::cout; using std::endl;

long fn1(long *p1, long *p2) {
    auto t = *p1;                        // Line 1
    *p1 = *p2;
    *p2 = t;
    return t;
}

long fn2(long &p1, long p2) {
    p1 *= 5;
    p2 *= 5;
    return p2;
}

int main (){
    long lng1 = 10;
    long lng2 = 20;
    long *p_lng1 = &lng1;
    long *p_lng2 = &lng2;
    cout << fn1(p_lng1, p_lng2) << endl; // Line 2
    cout << lng1 << endl;                   // Line 3
    cout << p_lng2 << endl;                 // Line 4
    cout << (p_lng1 == &lng1) << endl;      // Line 5

    long lng3 = 2;
    long lng4 = 4;
    cout << fn2(lng3, lng4) << endl;       // Line 6
    cout << lng3 << endl;                   // Line 7
    cout << lng4 << endl;                   // Line 8
}

Figure 1

1) For the program in Figure 1, what type is the variable t on Line 1?
   a) long
   b) long*
   c) bool
   d) size_t
   e) None of the above.

2) For the program in Figure 1, what output is given by Line 2?
   a) 0
   b) 10
   c) 20
   d) some address
   e) None of the above.
3) For the program in Figure 1, what output is given by Line 3?
   a) 0
   b) 10
   c) 20
   d) some address
   e) None of the above.

4) For the program in Figure 1, what output is given by Line 4?
   a) 0
   b) 10
   c) 20
   d) some address
   e) None of the above.

5) For the program in Figure 1, what output is given by Line 5?
   a) 0
   b) 10
   c) 20
   d) some address
   e) None of the above.

6) For the program in Figure 1, what output is given by Line 6?
   a) 0
   b) 10
   c) 20
   d) some address
   e) None of the above.

7) For the program in Figure 1, what output is given by Line 7?
   a) 0
   b) 10
   c) 20
   d) some address
   e) None of the above.

8) For the program in Figure 1, what output is given by Line 8?
   a) 0
   b) 10
   c) 20
   d) some address
   e) None of the above.
9) How many “children” (subnodes) can a single binary tree node have?
   a) 0
   b) 1
   c) 2
   d) All of the above
   e) None of the above

10) Which of the following are true statements regarding an insertion sort?
   a) For the data presented in the readings, it was faster than a selection sort.
   b) Had a BigO of O(n)
   c) Was competitive with the C++ built-in sort
   d) All of the above
   e) None of the above

11) Which of the following are differences between a C++ struct and class?
   a) only a class can have a constructor
   b) only a struct can have public data members
   c) by default a class’s data members are private, a struct’s public
   d) All of the above
   e) None of the above

12) Given the input stream cin, what does cin.ignore() do?
   a) ignores all further input from cin
   b) clears the contents of the cin buffer
   c) resets any errors that have occurred on cin
   d) All of the above
   e) None of the above

13) Which of the following are true about a friend function?
   a) It can only be used on constructors
   b) It gives the function access to private data members of a class/struct
   c) It cannot be used in templates
   d) All of the above.
   e) None of the above

14) Which of the following are true statements about the so-called “rule of three”
   a) it concerns the need for the copy constructor, destructor and operator=.
   b) is required for classes that utilize dynamic memory
   c) all 3 methods have defaults behaviors if not defined
   d) All of the above.
   e) None of the above.

15) Which of the follow are true about basic C++ arrays?
   a) they are fixed size.
   b) they do not store the size of the array
   c) the name of the array is a pointer type.
   d) All of the above
   e) None of the above
// Let's assume I got the includes correct

string fn1(vector<string>& v_s){
    sort(v_s.begin(), v_s.end());
    return v_s.front();
}

bool fn2(vector<string> &v_s, string s){
    bool result = false;
    auto i = lower_bound(v_s.begin(), v_s.end(), s); // Line 1
    if (*i != s){
        v_s.insert(i,s);
        result = true;
    }
    return result;
}

string fn3(vector<string> v_s, size_t i){
    string result=";
    if (i < v_s.size()){
        for(auto ele : v_s){ // Line 2
            result += ele[i];
        }
    }
    return result;
}

int main(){
    vector<string> v_s1 ={"jill", "fred", "jane", "bill"};
    cout << fn1(v_s1) << endl; // Line 3
    cout << v_s1.back() << endl; // Line 4

    vector<string> v_s2 ={"abc", "def", "ghi"};
    cout << fn2(v_s2, "ccc") << endl; // Line 5
    cout << v_s2.size() << endl; // Line 6
    cout << v_s2[1] << endl; // Line 7

    vector<string> v_s3 ={"mom", "dad", "bro", "sis"};
    cout << fn3(v_s3, 1) << endl; // Line 8
    cout << fn3(v_s3,6) << endl; // Line 9
}

Figure 2

16) For the program in Figure 2, what is the full type name for i on Line 1
a) vector::iterator
b) string::iterator
c) string*
d) vector
e) None of the above

17) For the program in Figure 2, what is the full type name for ele on Line 2
a) vector::iterator
b) string::iterator
c) string
d) string*
e) None of the above
18) For the program in Figure 2, what value is output on Line 3?
   a) jill
   b) fred
   c) jane
   d) bill
   e) None of the above

19) For the program in Figure 2, what value is output on Line 4.
   a) jill
   b) fred
   c) jane
   d) bill
   e) None of the above

20) For the program in Figure 2, what value is output on Line 5?
   a) 0
   b) 1
   c) true
   d) false
   e) None of the above

21) For the program in Figure 2, what value is output on Line 6?
   a) 4
   b) 3
   c) 2
   d) 0
   e) None of the above

22) For the program in Figure 2, what value is output on Line 7?
   a) abc
   b) def
   c) ghi
   d) ccc
   e) None of the above

23) For the program in Figure 2, what value is output on Line 8?
   a) mdbs
   b) momdadbrosis
   c) oari
   d) sisorbdadmom
   e) None of the above

24) For the program in Figure 2, what value is output on Line 9?
   a) mdbs
   b) momdadbrosis
   c) sisorbdadmom
   d) ccc
   e) None of the above
25) For the program in Figure 3, what value is output on Line 1?
   a) 0
   b) 1
   c) 2
   d) 3
   e) None of the above

```cpp
// data members public so we can test them
// assume includes are correct

class MyClass{
public:
    vector<char> v1_;  
    vector<char> v2_; 
    char c_ = '\0'; // null character

    MyClass()=default;
    MyClass(initializer_list<char> il); 
    bool fn1(char c); 
    char fn2(int); 
};

MyClass::MyClass(initializer_list<char> il){ 
    copy(il.begin(), il.end(), back_inserter(v1_)); 
}

bool MyClass::fn1(char c){
    bool result = false; 
    if ( (find(v1_.begin(), v1_.end(), c) == v1_.end()) 
        && (find(v2_.begin(), v2_.end(), c) == v2_.end()) 
    ){
        v1_.push_back(c);
        result = true; 
    }
    return result; 
}

char MyClass::fn2(int cnt){
    if ( cnt > v1_.size() )
        cnt = v1_.size(); 

    for(int i=cnt; i>0; --i){
        c_ = v1_.back(); 
        v1_.pop_back(); 
        v2_.push_back(c_); 
    }
    return c_; 
}

int main (){
    MyClass mc1{'a','b','c'}; 
    cout << mc1.v1_.size() << endl; // Line 1
    cout << mc1.fn1('d') << endl;  // Line 2
    cout << mc1.v1_.size() << endl; // Line 3
    cout << mc1.fn1('a') << endl;  // Line 4
    MyClass mc2{'x', 'f', 'r', 'p'}; 
    cout << mc2.fn2(2) << endl;    // Line 5
    cout << mc2.v1_.size() << endl; // Line 6
    cout << mc2.fn2(10) << endl;   // Line 7
    cout << mc2.v2_.size() << endl; // Line 8
}
```
26) For the program in Figure 3, what output is produced by Line 2?
   a) 0
   b) 1
   c) 2
   d) 3
   e) None of the above

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

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

29) For the program in Figure 3, what output is produced by Line 5?
   a) x
   b) f
   c) r
   d) p
   e) None of the above

30) For the program in Figure 3, what output is produced by Line 6?
   a) 1
   b) 2
   c) 3
   d) 4
   e) None of the above

31) For the program in Figure 3, what output is produced by Line 7?
   a) x
   b) f
   c) r
   d) p
   e) None of the above

32) For the program in Figure 3, what output is produced by Line 8?
   a) 0
   b) 1
   c) 2
   d) 3
   e) None of the above
// all data members public for testing
// assume proper includes

template<typename T>
class NewClass{
public:
   T* lptr_ = nullptr;
   size_t s_ = 0;

   NewClass()=default;
   NewClass(initializer_list<T> il);
   size_t fn1(T t);
   NewClass fn2(NewClass);
};

template<typename T>
NewClass<T>::NewClass(initializer_list<T> il){
   s_ = il.size();
   lptr_ = new T(s_);
   copy(il.begin(), il.end(), lptr_);
}

template<typename T>
size_t NewClass<T>::fn1(T val){
   size_t i=0;
   while( (i < s_) && (lptr_[i] != val) ){
      ++i;
   }
   return i;
}

NewClass<T> NewClass<T>::fn2(NewClass<T> nc){
   NewClass<T> temp;
   temp.s_ = s_ + nc.s_;
   temp.lptr_ = new T[temp.s_];
   copy(nc.lptr_, nc.lptr_+nc.s_, temp.lptr_);
   copy(lptr_, lptr_+s_, temp.lptr_+nc.s_);
   return temp;
}

int main (){  
   NewClass<long> nc1{12,13,14};
   NewClass<long> nc2{10,11};
   cout << nc1.s_ << endl;  // Line 1
   cout << nc2.fn1(13) << endl;  // Line 2
   cout << nc2.fn1(10) << endl;  // Line 3
   auto result = nc1.fn2(nc2);
   cout << result.lptr_[result.s_ - 1] << endl; // Line 4
}

Figure 4
33) For the program in Figure 4, what output is produced on Line 1?
   a) 0
   b) 1
   c) 2
   d) 3
   e) None of the above
34) For the program in Figure 4, what output is produced on Line 2?
   a) 0
   b) 1
   c) 2
   d) 3
   e) None of the above
35) For the program in Figure 4, what output is produced by Line 3?
   a) 0
   b) 1
   c) 2
   d) 3
   e) None of the above
36) For the program in Figure 4, what output is produced by Line 4?
   a) 14
   b) 13
   c) 12
   d) 11
   e) None of the above
37) For the program in Figure 5, what output is produced by Line 1?
   a) 0
   b) 1
   c) 2
   d) 3
   e) None of the above

38) For the program in Figure 5, what output is produced by Line 2?
   a) 100
   b) 200
   c) 300
   d) some address
   e) None of the above
39) For the program in Figure 5, what output is produced by Line 3?
   a) 100
   b) 200
   c) 300
   d) some address
   e) None of the above

40) For the program in Figure 5, what output is produced by Line 4?
   a) 0
   b) 1
   c) 2
   d) 3
   e) None of the above