

Spring Semester 15, Dr. Punch. Exam #2 (3/26), 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.

```

#include<iostream>
using std::cout; using std::endl;
#include<vector>
using std::vector;
#include<algorithm>
using std::sort;

double fun1(vector<long>& v){
    long s=0;
    for(auto i=v.begin(); i!=v.end(); i++) // Line 1
        s += *i;
    return s/v.size();
}

long fun2(vector<long> v){
    sort(v.begin(), v.end());
    long i = v.size()/2;
    return v[i];
}

void fun3(vector<long>&v){
    long t;
    for(int i=1; i<v.size(); i++){
        t = v[i-1];
        v[i-1]=v[i];
        v[i] = t;
    }
}

int main (){
    vector<long>v={1,3,5,7,9};
    cout << fun1(v) << endl; // Line 2
    cout << v[4] << endl; // Line 3
    cout << fun2(v) << endl; // Line 4
    cout << v[1] << endl; // Line 5
    fun3(v);
    cout << v[4] << endl; // Line 6
}

```

Figure 1

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

- 2) What output is produced by Line 2 in Figure 1?
- a) 1
 - b) 3
 - c) 5
 - d) 7
 - e) None of the above.
- 3) What output is produced by Line 3 in Figure 1?
- a) 1
 - b) 3
 - c) 5
 - d) 7
 - e) None of the above.
- 4) What output is produced by Line 4 in Figure 1?
- a) 1
 - b) 3
 - c) 5
 - d) 7
 - e) None of the above.
- 5) What output is produced by Line 5 in Figure 1?
- a) 1
 - b) 3
 - c) 5
 - d) 7
 - e) None of the above.
- 6) What output is produced by Line 6 in Figure 1?
- a) 1
 - b) 3
 - c) 5
 - d) 7
 - e) None of the above.

- 7) Which of the following clears the input buffer `cin` of all remaining input?
- a) `cin.clear`
 - b) `cin.ignore`
 - c) `cin.reset`
 - d) All of the above
 - e) None of the above
- 8) Which of the following are true about the `endl` statement?
- a) it is provided by the `ostream` header.
 - b) it flushes the buffer of the stream it is called on.
 - c) it inserts a line feed into the buffer.
 - d) All of the above
 - e) None of the above
- 9) Which of the following is true about an `ostringstream` variable `sout`?
- a) you can output to `sout` using `<<` operations.
 - b) it is a string so you can concatenate to it, for example `sout + "my string"`.
 - c) you **cannot** use format manipulators, for example `sout << setprecision(4)` is not allowed
 - d) All of the above.
 - e) None of the above.
- 10) What is meaning of the keyword `auto` in a type declaration?
- a) Can determine a variable's type at run time and compile 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 `long` or `string`.
 - d) All of the above
 - e) None of the above
- 11) Which of the following generic algorithms allows you to modify their "standard operation" by adding a function of your own design?
- a) `sort`
 - b) `accumulate`
 - c) `transform`
 - d) All of the above
 - e) None of the above
- 12) Which of the following are true about generic algorithms in general?
- a) they work on containers
 - b) they require iterators to work on containers
 - c) there are a lot of them, approximately 100
 - d) All of the above
 - e) None of the above
- 13) What does `=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 `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<utility>
using std::pair;
#include<string>
using std::string;

long fun1(map<long,long>&m, long l){
    long ans=0;
    for(auto i=m.begin(); i!=m.end(); i++){
        if ( (*i).first > l )
            ans += (*i).second;    // Line 1
    }
    return ans;
}

void fun2(map<long,long> &m, long k){
    if (m.count(k) == 1){
        m[k] = m[k] + k;
    }
    else
        m[k] = 0;
}

int main (){
    map<long, long> m={ {1,2}, {3,4}, {5,6}, {7,8} };
    cout << m.size() << endl;    // Line 2
    cout << m[5] << endl;        // Line 3
    cout << fun1(m, 3) << endl;   // Line 4
    fun2(m,9);
    cout << m.size() << endl;    // Line 5
}

```

Figure 2

14) For the program in Figure 2, what type is *i* in Line 1.

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

- 15) Still on Line 1 in Figure 2, what type is *i?
- a) pair<long, long>
 - b) pair
 - c) map::iterator
 - d) long
 - e) None of the above
- 16) Still on Line 1 in Figure 2, which of the following is an equivalent statement?
- a) ans += i.second;
 - b) ans += i->second;
 - c) ans += i->.second;
 - d) ans += second;
 - e) None of the above
- 17) For the program in Figure 2, give the output of Line 2.
- a) 3
 - b) 4
 - c) 5
 - d) 6
 - e) None of the above
- 18) For the program in Figure 2, give the output of Line 3.
- a) 3
 - b) 4
 - c) 5
 - d) 6
 - e) None of the above
- 19) For the program in Figure 2, give the output of Line 4.
- a) 12
 - b) 14
 - c) 16
 - d) 20
 - e) None of the above
- 20) For the program in Figure 2, give the output of Line 5.
- a) 3
 - b) 4
 - c) 5
 - d) 6
 - e) None of the above

```

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

class MyClass{
private:
    string s_;
    long v_cnt_;
    string v_ = "aeiou";
public:
    MyClass(string);
    string m1();
    friend ostream& operator<<(ostream&, MyClass&);
};

MyClass::MyClass(string s){
    s_ = s;
    v_cnt_ = 0;
    for (auto e : s_)           // Line 1
        if (v_.find(e) != string::npos) // Line 2
            v_cnt_++;           // Line 3
}

string MyClass::m1(){
    string ans = "";
    for (auto e : s_)
        if (v_.find(e) == string::npos)
            ans += e;
    return ans;
}

ostream& operator<<(ostream &out, MyClass &c){
    out << c.s_ << "," << c.v_cnt_;
    return out;
}

int main (){
    MyClass mc("abcdefgh");
    cout << mc << endl;           // Line 4
    cout << mc.m1() << endl;      // Line 5
}

```

Figure 3

21) For the program in Figure 3, what type is e on Line 1?

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

- 22) For the program in Figure 3, what is signified by `string::npos` on Line 2?
- a) no such position in the string
 - b) the first position of the string
 - c) the last position of the string
 - d) a count of the positions in string
 - e) None of the above
- 23) For the program in Figure 3, which of the following is equivalent for Line 3?
- a) `(this.v_cnt_++)`
 - b) `(this->v_cnt_++)`
 - c) `(self.v_cnt_++)`
 - d) `this.(*v_cnt_++)`
 - e) None of the above
- 24) For the program shown in Figure 3, what output is produced by Line 4?
- a) `abcdefg`
 - b) `abcdefg, 5`
 - c) `abcdefg, 2`
 - d) `bcd fgh`
 - e) None of the above
- 25) For the program shown in Figure 3, what output is produced by Line 5?
- a) `abcdefg`
 - b) `abcdefg, 5`
 - c) `bcd fgh, 2`
 - d) `empty string`
 - e) None of the above


```

// lots of includes, assume everything is there
class AClass{
private:
    map<long, vector<long>> m_;

public:
    AClass() =default;
    long m1(long);
    long m2(long);
    friend ostream& operator<<(ostream&, AClass&);
};

long AClass::m1(long n){
    long cnt=0;
    for(int i=1; i<n; i++){
        if (n % i == 0){
            m_[n].push_back(i);
            cnt++;
        }
    }
    return cnt;
}

long AClass::m2(long n){
    long ans=0;
    for (auto e : m_[n])
        if (e % 2 == 0)
            ans++;
    return ans;
}

string pair_to_string(const pair<long, vector<long>>&p){
    string ans = to_string(p.first) + ": ";
    for(auto e : p.second)
        ans += to_string(e) + ",";
    return ans.substr(0,ans.size()-1);
}

ostream& operator<<(ostream& out, AClass &ac){
    transform(ac.m_.begin(), ac.m_.end(), ostream_iterator<string>(out, "\n"),
        pair_to_string);
    return out;
}

int main (){
    AClass ac;
    cout << ac.m1(12) << endl;    // Line 1
    cout << ac << endl;          // Line 2
    cout << ac.m2(12) << endl;    // Line 3
    cout << ac.m2(24) << endl;    // Line 4
    cout << ac << endl;          // Line 5, be careful!
}

```

Figure 4

- 26) For the program in Figure 4, what value is printed by Line 1?
- a) 2
 - b) 3
 - c) 4
 - d) 5
 - e) None of the above
- 27) For the program in Figure 4, what value is printed by Line 2?
- a) 12
 - b) 12: 1, 2, 3, 4, 5
 - c) 12: 2, 3, 4, 6
 - d) 12: 1, 2, 3, 4, 6
 - e) None of the above
- 28) For the program in Figure 4, what value is printed by Line 3?
- a) 2
 - b) 3
 - c) 4
 - d) 5
 - e) None of the above
- 29) For the program in Figure 4, what value is printed by Line 4?
- a) 2
 - b) 3
 - c) 0
 - d) 1
 - e) None of the above
- 30) For the program in Figure 4, what value is printed by Line 5?
- a) 12
 - b) 12: 1, 2, 3, 4, 5
 - c) 12: 2, 3, 4, 6
 - d) 12: 1, 2, 3, 4, 6
 - e) None of the above