Name [redacted]

CS 31 Lectures 2 and 3
Fall 2009
Midterm Exam
November 4, 2009

Problem #	Possible Points	Actual Points
1	10	10
2	10	10
3	10	10
4	10	10
5	10	10
6	10	10
7	15	~~10~~ → 15
8	10	10
9	15	2
TOTAL	100	~~82~~ → 87

STUDENT ID #: [redacted]

SIGNATURE:

ENROLLED IN LI ne): 1 [✓ 2] 3

CLOSED BOOK, TWO 8.5"x11" SHEETS
NO ELECTRONIC DEVICES

ENJOY!

1. [10 points in all]

a. [5 points] Consider the following program:

```
#include "std_lib_facilities.h"

bool isSorted(const vector<int>& v) {3, 2, 1}
{
    for (int k = 0; k != v.size(); k++)
        if (v[k] > v[k+1])
            return false;      ← will cause error
    return true;
}

int main()
{
    vector<int> w;
    w.push_back(10);
    w.push_back(20);
    w.push_back(20);
    w.push_back(30);
    w.push_back(40);
    if (isSorted(w))
        cout << "The vector is sorted" << endl;
    else
        cout << "The vector is out of order" << endl;
}
```

Which of these statements is true about the program? (Circle a, b, or c, and then write something as indicated below):

a. It is a well-formed C++ program with well-defined behavior.

[✓ b.] It has no error that would prevent compilation, but when executed has undefined behavior or does something to cause it to end prematurely.

c. It has at least one error that will prevent compilation from succeeding.

If you circled a, show the output the program writes; if you circled b or c, briefly state the problem.

It will run out of bounds when k = v.size() - 1 because v[k+1] will be v[v.size()] which is out of bounds.

b. [5 points]
If you are enrolled in lecture 1 or 2 (Smallberg): In a brief English sentence, state the primary purpose of the C++ compiler.
If you are enrolled in lecture 3 (Rohr): State the definition (used in this class) of an electronic digital computer.

A C++ compiler translates C++ code into machine code so the computer is able to run it

2. [10 points] What is the output produced by the following program fragment?

```
int n = 6;
while (n/2 != 7   &&   n < 16)
{       n != 14, 15 & n<16
      cout << n << endl;
      n++;
      if (n % 2 == 0)
            n += 3;
}
cout << "Finish: n = " << n << endl;
```

print out	number trace
6	7
7	8
11	11
	12
Finish 15	15

```
6
7
11
Finish: n = 15
```

print out	number trace
6	6
7	7
11	8
15	11
	12
	15

3. [10 points] First, write a single statement that creates an empty vector of doubles. Name the vector b. Then, write a small number of statements that causes b to contain 200 elements, where the element at the beginning of the vector has the value 1, and each of the other 199 elements has a value that is 1.04 times the value before it (so that the element after the one with the value 1 has the value 1.04, the one after that has the value of 1.04^2, etc.).

```
vector<double> b;
b.push_back(1);
for(int i=0; i<199; i++)
    b.push_back(b[i]*1.04);
```

4. [10 points] The following program is supposed to read in ten strings. For each string read in, the program should print how many vowels are in that string, where (only) the letters A, E, I, O, and U in either case are considered to be vowels.

```
#include "std_lib_facilities.h"

bool contains(string s, char ch)
{
     for (int k = 0; k != s.size(); k++)
          if (s[k] == ch)
               return true;
     return false;
}

int main()
{
     string text;
     int vowelCount = 0;
     for (int n = 0; n < 10; n++)
     {
          cout << "Enter a string: ";
          getline(cin, text);
          for (int k = 0; k != text.size(); k++)
               if (contains("AEIOUaeiou", text[k]))
                    vowelCount++;
          cout << "There are " << vowelCount << " vowels"
               << endl;
     }
}
```

(continued on next page)

4. (continued) A sample transcript *should* start out

```
Enter a string: Remember, remember the fifth of November
There are 12 vowels
Enter a string: Gunpowder, treason and plot
There are 8 vowels
... etc. ...
```

The program builds fine, but does not work as intended. If the first two strings entered as input to the program are the two in the sample transcript on the previous page, what are the first lines the program actually writes (other than the `Enter a string:` prompts)? (If the program worked as intended, you'd answer

```
There are 12 vowels
There are 8 vowels
```

but the program doesn't work as intended.)

There are 12 vowels

There are 20 vowels

How can the program be changed to fix the problem? (You may either clearly indicate the change below, or rewrite a portion of the code.)

✓

```
#include "std_lib_facilities.h"

bool contains(string s, char ch)
{
     for (int k = 0; k != s.size(); k++)
          if (s[k] == ch)
               return true;
     return false;
}

int main()
{
     string text;
     int vowelCount = 0;
     for (int n = 0; n < 10; n++)
     {
          cout << "Enter a string: ";
          getline(cin, text);
          for (int k = 0; k != text.size(); k++)
               if (contains("AEIOUaeiou", text[k]))
                    vowelCount++;
          cout << "There are " << vowelCount << " vowels"
               << endl;
     }
}
```

vowelCount =0; before 2nd for loop

5. [10 points] Convert this switch statement to code that produces exactly the same output without using a switch statement. (We'll give you a break by reminding you to read the switch statement carefully.)

```
char insurance;
 ... assume some intervening code here that gives insurance a value ...
switch (insurance)
{
  case 'F':
     cout << "fire" << endl;
     break;
  case 'E':
     cout << "earthquake" << endl;
  case 'C':
  case 'T':
     cout << "car/truck" << endl;
     break;
  default:
     cout << "other" << endl;
     break;
}
```

```
if (insurance == 'F')
   cout << "fire" << endl;
else if (insurance == 'E' || insurance == 'C' || insurance == 'T')
   {
   if (insurance == 'E')
      cout << "earthquake" << endl;
   cout << "car/ truck" << endl;
   }
 else
   cout << "other" << endl;
```

✓

6. [10 points] Find and correct all of the errors (both syntactic and semantic) in the sum function below. The sum function, once corrected, should return the sum of *all the doubles* in the vector passed in to it. Do not completely rewrite the function; just *clearly* correct the bugs. Do not use any loop other than a while loop.

```
#include "std_lib_facilities.h"
          ← because it returns the sum
double ~~void~~ sum(const vector<double>& u)
{
    int n = u.size();   v.size()-1;  ← so it starts in bounds
double ~~int s;~~  s = 0;  ← so it doesn't initiate @ junk value

    ~~while (n > 0);~~ while (n >= 0)   ← no ; with while statement
    {                       ↖ so it includes index 0
      s += u[n];
      n--;
    }
    ~~Return s;~~
}   return s;   ← no cap on return
```

so it is capable of being the sum value since vector contains doubles

+7

Now consider this main routine:

```
int main()
{
    vector<double> scores;
    scores.push_back(1);        11
    scores.push_back(10);
    scores.push_back(4.8);      15.8
    scores.push_back(0);        15.8
    scores.push_back(3);        18.8
    cout << sum(scores);
}
```

+3

Assuming sum has been corrected, what will this main routine print? (Circle one choice)

(a) 5 (b) 17.8 (c) 18 [✓ (d) 18.8] (e) 19 (f) some other well-defined number

(g) unpredictable, since the program has undefined behavior

7. [15 points] The integer 13 is the sum of two integer squares, since 13 = $3^2 + 2^2$. Write a function named `squaresums` that takes an int parameter named `n` and returns void. For all positive s, j, and k such that s <= n and s = $j^2 + k^2$, the function must write exactly one line with one of the triples (s, j, k) or (s, k, j). (When n < 2, this means there will be no output.) It can write these lines of triples in whatever order you choose. For example, if the function is called with the argument 14, it could write

```
2  1  1
10  1  3          this could instead appear as 10   3   1
8  2  2
13  2  3          this could instead appear as 13   3   2
5  1  2           this could instead appear as 5   2   1
```

$s=j^2+k^2$

These five lines could instead be in any other order (e.g., 2 1 1 doesn't have to be the first line, as long as it is one of the five lines).

$j^2+k^2 \leq n$

$j^2 < n$

$k^2 < n$

Write the `squaresums` function using correct C++ syntax and appropriate style. (You do not have to write a main routine or #include directive.)

```
void    squaresums( int  n)
    {
     for (int j= 1;(j*j)< n; j++)    //use j*j instead of j
        for(int k=j;(k*k)<n; k++)    // because s cannot ~~possibly be~~
          {                          // ≤n if one of its components
           int s=(j*j )+(k*k);       // is not less than n
           if( s ≤ n)                // (same for k)
                                     // saves time
              cout<< s <<" "<< j <<" " << k << endl;
          }
    }
```

10, 1, 3

10, 3, 1

Testing Zone value= 14

j=1

k=1

p 2 1 1

k=2

p 5 1 2

k=3

p 10 1 3

k=4

j=2

k=2

p 8 2 2

k=3

p 13 2 3

k=4

j=3

k=3

k=4

8. [10 points] What is the fifth digit of your UCLA student ID number? 6
What output is produced by the following program if its input is the fifth digit of your UCLA student ID number?

flu	poo	output
6	30	6 30
6	30	6 30
6	30	6
7	30	7
8	30	flu

```
#include "std_lib_facilities.h"

void stew(int flu, int who);
void coup(int& poo);
void through(string blue);

int main()
{
    int flu;
    cin >> flu;       // you enter the fifth digit of
                      // your student id
    int poo = 30;
    stew(flu, poo);
    stew(flu, poo);
    coup(flu);
    coup(flu);
    through("flu");
}

void stew(int flu, int who)
{
    cout << flu << " " << who << endl;
    flu++;
    who = 20;
}

void coup(int& poo)
{
    cout << poo << endl;
    poo++;
}

void through(string blue)
{
    cout << blue << endl;
}
```

```
6 30
6 30
6
7
flu
```

9. [15 points] Write a function named `hasMajority` that takes two parameters:

a reference to a constant vector of integers named `data`
a reference to integer named `value`

The function returns a bool. If there exists an integer that appears in *more than* half of the elements of the vector, the function sets `value` to that integer and returns true; otherwise it leaves `value` unchanged and returns `false`. As an example, for this main routine:

```
int main()
{
        vector<int> nums;
        nums.push_back(20);
        nums.push_back(10);

        int val = 99;

        if (hasMajority(nums, val))
                cout << "Has a majority";
        else
                cout << "No majority";
        cout << ", val is " << val << endl;

        nums.push_back(10);
        nums.push_back(30);
        nums.push_back(10);

        if (hasMajority(nums, val))
                cout << "Has a majority";
        else
                cout << "No majority";
        cout << ", val is " << val << endl;
}
```

the output would be:

```
No majority, val is 99
Has a majority, val is 10
```

In the first case, the `nums` vector has one occurrence of 20 and one of 10, so since no value occurs more than 50% of the time, the function returns false and leaves `val` unchanged. In the second case, the vector of 5 elements has 3 occurrences of 10, which is more than half of them.

(continued on next page)

9. (continued) Write your **`hasMajority`** function here. (You do not have to write a main routine or #include directive.)

```
bool    hasMajority (const  vector<int>& data, int& value)
    {
    int  counter = 0;
    for(int i = 0; i < data.size(); i++)
        if (data[i] == value)
            counter ++;
    return (2*counter) > data.size();
    }
```

solved   Wrong problem, we're not checking if

value is in majority.

2/15