⚠ This quiz has been regraded; your new score reflects 2 questions that were affected.

Midterm Exam

Due Feb 8 at 5:50pm **Points** 100 **Questions** 17
Available Feb 8 at 4pm - Mar 20 at 11:59pm **Time Limit** 110 Minutes

Instructions

Section 102.01 of UCLA's **UCLA Student Conduct Code (https://www.deanofstudents.ucla.edu/portals/16/documents/uclacodeofconduct_rev030416.pdf)** prohibits all forms of academic misconduct or research misconduct, including, but not limited to, cheating, fabrication or falsification, plagiarism, multiple submissions, facilitating academic dishonesty, coercion regarding grading or evaluation of coursework, or unauthorized collaboration. By submitting this exam for grading, I acknowledge these terms. I declare that this work is solely my own and that I have not communicated with anyone other than the instructor and proctors in any way during the exam.

This quiz was locked Mar 20 at 11:59pm.

Attempt History

	Attempt	Time	Score	Regraded
LATEST	Attempt 1	109 minutes	85.5 out of 100	110 out of 100

⚠ Correct answers are no longer available.

Score for this quiz: **110** out of 100
Submitted Feb 8 at 5:49pm
This attempt took 109 minutes.

The following program code will be used for the next 3 questions:

```
// Electoral College
#include <iostream>
using namespace std;
```

```
int main()
{

   int totalVoters;
   int republicanVotes, democraticVotes;

   cout << "How many Electoral College electors are there? ";
   cin >> totalVoters;
   cout << "How many of these electors voted for the Republican party
candidate? ";
   cin >> republicanVotes;
   cout << "How many of these electors voted for the Democratic party
candidate? ";
   cin >> democraticVotes;

   double pctRepub = ( 100.0 * republicanVotes ) / totalVoters;
   double pctDemoc = ( 100.0 * democraticVotes ) / totalVoters;

   cout.setf(ios::fixed);  // see pp. 30-31 in Savitch book
   cout.precision(1);
   cout << endl;

   cout << pctRepub << "% of the " << totalVoters << " electors voted for
the Republican party candidate." << endl;
   cout << pctDemoc << "% of the " << totalVoters << " electors voted for
the Democratic party candidate." << endl;
   cout << endl;

   if (pctRepub > pctDemoc)
   {
      cout << "Looks like the Republican party candidate won!" << endl;
   }
   if (pctDemoc > pctRepub)
   {
      cout << "Looks like the Democratic party candidate won!" << endl;
   }

   if (republicanVotes + democraticVotes != totalVoters)
   {
      cout << "Looks like some of these data values don't make sense." <<
endl;
   }
```

```
    return(0);
}
```

Question 1

1 / 1 pts

Which of the following input provided to the program would generate the messages:

10.0% of the 10 electors voted for the Republican party candidate.
10.0% of the 10 electors voted for the Democratic party candidate.
Looks like some of these data values don't make sense.

- (●) 10 1 1
- ( ) 20 2 2
- ( ) 1 1 10
- ( ) 100 10 10
- ( ) 10 100 -10

Your answer is correct

Question 2

1 / 1 pts

Which of the following input provided to the program would generate the messages:

51.0% of the 100 electors voted for the Republican candidate.
20.0% of the 100 electors voted for the Democratic candidate.
Looks like the Republican candidate won!
Looks like some of these values don't make sense.

- 100 51 20
- 20 51 100
- 51 100 20
- 51 20 100
- 20 100 51

Correct.

Question 3

Original Score: 0 / 2 pts Regraded Score: 2 / 2 pts

This question has been regraded.

This program will skip all input requests if the user enters a character "k" when the program prompts, "How many Electoral College electors are there? "

Which statement causes this input-skipping problem (think critically - think about why the problem occurs)?

-
-
- cin >> democraticVotes;
- cout << "How many of these electors voted for the Democratic party candidate? ";
- cin >> republicanVotes;

cout << "How many of these electors voted for the Republican party candidate? ";

cin >> totalVoters;

cout << "How many Electoral College electors are there? ";

Question 4

2 / 2 pts

What gets printed by the following code?

```
cout << 18 + 2 * 3 - 1;
```

- 23
- 22
- 59
- 18 + 2 * 3 - 1
- 40

Question 5

2 / 2 pts

What gets printed by the following code?

```
cout << 37 / 6 - 1;
```

- 5
- 5.00000

- 5.166667
- 37 / 6 - 1
- 6

Question 6

2 / 2 pts

What will be printed by the following code?

```
int i = 50;

if (i < 30)
{
  cout << “i < 30!” << endl;
  if (i < 20)
  {
    cout << “i < 20!” << endl;
  }
}
else
{
  cout << “yay!” << endl;
}
```

- i < 30!
- i < 30! i < 20!
- yay!
- i < 20! yay!
- i < 20!

Question 7

Original Score: 0 / 2.5 pts Regraded Score: 2.5 / 2.5 pts

This question has been regraded.

After executing the statement:

bool b = (false && true) || ((false || true);

the boolean b will have the value

- (●) true
- ( ) false
- ( ) Actually, a compile error will occur that will prevent this statement from building properly.
- ( ) Actually, a runtime error will occur that will prevent this statement from executing properly.

Question 8

2.5 / 2.5 pts

After executing the statement:

bool b = (false || true) && (false && true);

the boolean b will have the value

- ( ) true
- (●) false

Actually, a compile error will occur that will prevent this statement from building properly.

Actually, a runtime error will occur that will prevent this statement from executing properly.

Question 9

2.5 / 2.5 pts

Select the statement that declares a function foo that returns an integer value and accepts a single floating point number argument.

- (selected) int foo(double f);
- double foo(int i);
- void foo(double f);
- int foo(double f, double g);
- integer foo(double f);

Question 10

2.5 / 2.5 pts

Patty the Programmer is writing code to implement the function declared as:

```
void foo( int & i, int & j, int k );
```

She wants to change both the values of i and k to -1. Which of the following statements will complete this task?

- (●) i = -1; k = -1;
- ( ) &i = 1; &k = -1;
- ( ) &i = -1; k = -1;
- ( ) i = -1; &k = -1;

Question 11

2.5 / 2.5 pts

Patty the Programmer is writing code in her main( ) that says:

```
bar( i, j, k );
```

How is the parameter i being passed to the function bar?

- (●) Not enough information has been provided to determine the parameter passing scheme
- ( ) pass-by-value is being used
- ( ) pass-by-reference is being used
- ( ) pass-by-constant-reference is being used
- ( ) another parameter passing scheme not mentioned here is being used

Question 12

5 / 5 pts

Given the function definition, which of the following are correct?

```
int func(int n, double d)
{
   int j = n;
```

```
    double sum = 0;
    while( j >= 0)
    {
       sum += d;
       -j;
    }
    return sum;
}
```

With arguments 7 and 2.0

- returns 7!
- returns 7*2
- There is a syntax error in the program so it won't run.
- returns 7+2
- It compiles but computes none of these.

Question 13

2.5 / 2.5 pts

Which of the following loop statements is guaranteed to iterate the body of the loop at least once?

- for (initialize; test; update) body;
- while(control) body;
- do body while(control);
- all of the above
- none of the above

Question 14

20 / 20 pts

Please implement a function that accepts a string parameter as well as a boolean parameter. Your function should return the number of times it finds characters inside the string that are A's, B's or C's. When all three characters are found in the string and all the A's occur before any B's and all the B's occur before any C's in the string, set the boolean argument to true. Otherwise, set that boolean argument to false. If there are not any A's, B's or C's found, set that boolean argument to false. Return -1 if the string argument is the empty string. The declaration for this function will be:

int allAsBeforeBsBeforeCs(string s, bool & foundAllThree);

Here are some examples of how a main routine could test this function:

```
bool b = false;

assert( allAsBeforeBsBeforeCs( "", b ) == -1 );
// string argument is the empty string
assert( b == false );

assert( allAsBeforeBsBeforeCs( "99912000", b ) == 0 );
// zero A's B's or C's found
assert( b == false );
// no A's B's or C's found


assert( allAsBeforeBsBeforeCs( "1CC2BB3AA", b  ) == 6 );
// six A's B's and C's found
assert( b == false );
// all A's do not come before all B's before all C's

assert( allAsBeforeBsBeforeCs( "123A456B789C", b ) == 3 );
// three A's B's and C's found
assert( b == true );
// all A's come before B's come before C's

assert( allAsBeforeBsBeforeCs( "1CC2BB3AABC", b ) == 8 );
// eight A's B's and C's found
assert( b == false );
// A's are not coming before all B's before all C's
```

Write your allAsBeforeBsBeforeCs function in the text box below. You do not have to write a main routine or #include directives.

Your Answer:

```
int allAsBeforeBsBeforeCs( string s, bool & foundAllThree ) {
    int sum = 0;
    bool a = false, b = false, c = false;

    if (size(s) == 0) return -1;

    for (int i = 0; i < size(s); i++) {
        if (s[i] == 'A' || s[i] == 'B' || s[i] == 'C') sum++;
    }

    for (int i = 0; i < size(s); i++) {
        if (s[i] == 'A') {
            a = true;
            if (b == true || c == true) {
                foundAllThree = false;
                break;
            }
        }
        if (s[i] == 'B') {
            b = true;
            if (a == false || c == true) {
                foundAllThree = false;
                break;
            }
        }
        if (s[i] == 'C') {
            c = true;
            if (a == false && b == false) {
                foundAllThree = false;
                break;
            }
        }
        foundAllThree = true;
    }
```

```
    if (a == false || b == false || c == false) foundAllThree = false;


    if (sum == 0) foundAllThree = false;
    return sum;
}
```

Question 15

10 / 20 pts

Please implement a function that accepts a string parameter, two character parameters as well as a boolean parameter. If the string parameter is the empty string, set the boolean argument to false. Otherwise, your function should return the number of times it finds each character parameter inside the string. It these two counts are equal, set the boolean argument to true. Otherwise, set that boolean argument to false. The declaration for this function will be:

void hasThatManyOfThis(string s, char thatCharacter, char thisCharacter, bool & answer);

Here are some examples of how a main routine could test this function:

```
bool b = false;

hasThatManyOfThis( "", 'Z', 'A', b );
// the string parameter is the empty string
assert( b == false );

hasThatManyOfThis( "HelloWorld", 'Z', 'A', b );
// zero A's and zero Z's found in the string
assert( b == true );

hasThatManyOfThis( "HelloWorld", 'H', 'W', b );
// one H's and one W's found in the string
assert( b == true );

hasThatManyOfThis( "HeolloWorld", 'l', 'o', b );
// three l's and three o's found in the string
assert( b == true );

hasThatManyOfThis( "HeolloWorldooo", 'l', 'o', b );
// three l's and six o's found in the string
assert( b == false );
```

Write your hasThatManyOfThis function in the text box below. You do not have to write a main routine or #include directives.

Your Answer:

```
int hasThatManyOfThis( string s, char thatCharacter, char thisCharacter,
bool & answer ){
    int sumThatCharacter = 0;
    int sumThisCharacter = 0;

    for (int i = 0; i < size(s); i++) {
        if (s[i] == thatCharacter) sumThatCharacter++;
        if (s[i] == thisCharacter) sumThisCharacter++;
    }

    if (size(s) == 0) answer = false;
    else if (sumThatCharacter == sumThisCharacter) answer = true;
    else answer = false;

    return sumThatCharacter + sumThisCharacter;
}
```

syntax

Question 16

15 / 15 pts

In the text box below, please convert the while loop below to code that produces the same output but uses a do-while loop without any other kind of loop. Assume I and j were declared previously and initialized to any numbers at the beginning of the main function before the while loop. Your loop must be able to handle i and j of any numerical values.

```
 while( i < j )
 {
    cin >> j;
    cin >> i;
```

```
    cout << "i = " << i << " j = " << j << endl;
  }
```

Your Answer:

```
do {
   if (j <= i) break;
   cin >> j;
   cin >> i;
   cout << "i = " << i << " j = " << j << endl;
} while( i < j );
```

Question 17

15 / 15 pts

In the text box below, please convert the do-while loop shown below to code that produces exactly the same output but uses a for loop without any other kind of loop. Assume I and j were declared previously and initialized to any numbers. Your loop must be able to handle i and j of any numerical values.

```
  do
  {
     cin >> j;
     cin >> i;
     cout << "i = " << i << " j = " << j << endl;
  } while ( j > i );
```

Your Answer:

```
for (;;) {
   cin >> j;
   cin >> i;
   cout << "i = " << i << " j = " << j << endl;
   if (j <= i) break;
}
```

Quiz Score: **110** out of 100

This quiz score has been manually adjusted by +20.0 points.