Winter 2018 – CS 31
Midterm Exam Version C

Problem #	Possible Points	Actual Points
1	5	
2	5	
3	5	
4	10	
5	10	
6	10	
7	10	
8	10	
9	10	
10	10	
11	16	
TOTAL	100	

FIRST NAME: ______________________________

LAST NAME: ______________________________

STUDENT ID #: ______________________________

SIGNATURE: ______________________________

CLOSED BOOK

ONE 8.5"x11" SHEET OF NOTES ALLOWED

NO ELECTRONIC DEVICES

GOOD LUCK ONE AND ALL!

1. **[5 points]**
   Choose the single best answer to each question and circle it in this table. If you believe a question to be ambiguous or erroneous, place a * next to the letter of the question and then add a comment next to the question.

A	B	C	D	E
A B C D E	A B C D E	A B C D E	A B C D E	TRUE FALSE

A. What will be printed by the following code?

```
int i = 20;
if (i == 12)
{
  cout << “i is 12!” << endl;
}
else if (i < 20)
{
  cout << “i < 20!” << endl;
}
cout << “yay!” << endl;
```

```
A. i is 12!   B. i is 12!   C. yay!   D. i < 20!   E. i is 12!
                 i < 20!                  yay!          yay!
```

B. What will be printed by the following code?

```
int i = 12;
if (i == 12)
{
  cout << “i is 12!” << endl;
}
if (i < 20)
{
  cout << “i < 20!” << endl;
}
else
{
  cout << “yay!” << endl;
}
```

```
A. i is 12!   B. i is 12!   C. yay!   D. i < 20!   E. is 12!
                 i < 20!                  yay!          yay!
```

C. What will be printed by the following code?

```
int i = 20;
if (i < 30)
{
  cout << “i < 30!” << endl;

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

```
A. 20 < 30!   B. i < 30!    C. yay!    D. i < 20!    E. i < 30!
                 i < 20!                  yay!
```

D. What will be printed by the following code?

```
int i = 20;
if (i < 20)
{
  cout << “i < 20!” << endl;

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

```
A. i < 30!    B. i < 20!    C. yay!    D. i < 20!    E. i < 20!
                 i < 30!                  yay!
```

E. Within a C++ program, the body of a `do while` loop may never execute. TRUE or FALSE

2. [5 points] Cross out all the cout statements that do not print exactly what is printed by the first cout statement below:

```
cout << ‘5’ << endl;


cout << "5" << endl;

cout << (9+1)/2 << endl;

cout << (9.0+1)/2 << endl;

cout << (9.0+1)/2.0 << endl;

cout << 9 + 1 / 2 << endl;

cout << 11/2 << endl;

cout << 11/2.0 << endl;

cout << 11.0/2 << endl;
```

3. [5 points] In the box below, show what is printed by the following program.

```
#include <iostream>
using namespace std;

void swap(int a, int & b)
{
   int temp = a;
   a = b;
   b = temp;
}

int main()
{
   int a = 14;
   int b = 12;
   int c = 15;

   swap(a,b);
   swap(b,c);
   cout << "a=" a << " b=" << b << " c=" << c << endl;
}
```

4. [10 points] Below is the header for a C++ function named zeroOne; in this question, you will need to write just the body of this function. The function should read a string. If the string entered is either “0” or “1”, then that entered string should be assigned to parameter s, and the function should end. If the user enters any other string value, the function should keep asking the user to re-enter a valid string until they type either “0” or “1”. Once you complete the function, use the supplied main program shown below to call your zeroOne function using the variable value.

```
#include <iostream>
#include <string>
using namespace std;

void zeroOne (string& s)
{





















}

int main()
{
   string value;
   // call zeroOne below using the variable value



   cout << "value = " << value << endl;
   return( 0 );
}
```

5. [10 points] Write a second version of zeroOne that uses the function return mechanism to return the string value entered (instead of using the reference parameter). No parameter will be necessary since you are using the return value instead. Name this second function zeroOne2. Once you complete the function, use the supplied main program shown below to call your zeroOne2 function using the variable value.

```
#include <iostream>
#include <string>
using namespace std;
```

```
int main()
{
   string value;
   // call zeroOne2 below using the variable value


   cout << "value = " << value << endl;
   return( 0 );
}
```

6. [10 points] In the box below, declare and initialize an array called numbers with the following 6 values: 10, 5, 0, 1, 7, 8.

Assume the array numbers has been initialized as stated above, show its contents after executing the following loop:

```
for (int i=1; i <= 6; i++)
{
  numbers[i-1] = i * i - numbers[i-1] + 1;
}
```

INDEX	[0]	[1]	[2]	[3]	[4]	[5]	[6]
VALUE							

7. [10 points]

Choose the single best answer to each question and circle it in this table. If you believe a question to be ambiguous or erroneous, place a * next to the letter of the question and then add a comment next to the question.

A	B	C	D	E
TRUE	TRUE	TRUE	TRUE	TRUE
FALSE	FALSE	FALSE	FALSE	FALSE

A. In a C++ program, keywords (like `if` or `switch`) cannot be used by programmers as the name of a variable or function.

B. Given the program:

```
#include <iostream>
using namespace std;

int main( )
{
    int 2018UclaCS31( 0 );
    while( int i = 0; i < 2018UclaCS31; i++ )
         cout << i << endl;
    return( 0 );
}
```

C++ will be able to build and run this code without any errors.

C. Given the program:

```
#include <iostream>
using namespace std;

int main( )
{
    char aa( "aa" );
    cout << aa;
    return( 0 );
}
```

C++ will be able to build and run this code without any errors.

D. Given the program:

```
#include <iostream>
#include <string>
using namespace std;

int main( )
{
    string empty( '' );
    string space( ' ' );

    if (empty != space)
      cout << 'These strings are not equal!' << endl;
    return( 0 );
}
```

C++ will be able to build and run this code without any errors.

E. Given the program:

```
#include <iostream>
#include <string>
using namespace std;

int main( )
{
    string s;
    cin  >> s;
    cout << s << endl;
    return( 0 );
}
```

If the running program is supplied the input: `Humpty Dumpty!`
the program will output: `Humpty Dumpty!`

8. [10 points] Consider the following code:

```
#include <iostream>
using namespace std;

void do_calc( int n1, int & n2 );

int main( )
{
  int num1 = 15, num2 = 16;
  do_calc( num2, num1 );
  cout << num2 << " | " << num1 << endl;   // line A
  do_calc( num1, num2 );                   // line B
  cout << num2 << " | " << num1 << endl;   // line C
  return( 0 );
}

void do_calc( int n1, int & n2 );
{
  int val = 1;
  n1 = n1 * 4;
  n2 = n2 + 4;
  val++;                                   // line D
  cout << n2 << " | " << n1 << endl;
}
```

What exactly is output to the screen at the line commented line A?

What exactly is output to the screen at the line commented line C?

What will be held in the variable `val` after the line commented line D executes during the second call to the function `do_calc` from the line commented line B?

9. [10 points] Using the blank page that follows this question, convert this switch statement to code that produces exactly the same output but does not use a switch statement.

(Please read the code carefully!)

```
char letter;
// a bunch of code goes here that gives letter a value...
switch (letter)
{
  case 'J':
     cout << "January" << endl;
     break;
  case 'S':
     cout << "September" << endl;
  case 'O':
  case 'o':
     cout << "October" << endl;
     break;
  default:
     cout << "Unknown month!" << endl;
     break;
}
```

9. **Print your answer below**

10. [10 points] Pasted below is a portion of your instructor's solution to Project 2 that follows all the input processing and data verification steps. Please make the necessary changes so that everyone over 70 cannot donate, no matter their gender.

```
        /* Enforcing the different parts of the assignment table */
        if (weight < 110 || age < 16)
        {
            okayToGiveBlood = false;
        }
        else if (gender == MALE_GENDER && (age >= 16 && age <= 22) &&
                     height >= 60 && weight >= 110)
        {
            okayToGiveBlood = true;
        }
        else if (gender == MALE_GENDER && age > 22 && weight >= 110)
        {
            okayToGiveBlood = true;
        }
        else if (gender == MALE_GENDER)
        {
            okayToGiveBlood = false;
        }
        else if (gender == FEMALE_GENDER && (age >= 16 && age <= 22))
        {
            if (height < 58)
            {
                okayToGiveBlood = false;
            }
            else if (height == 58)
            {
                if (weight >= 146)
                    okayToGiveBlood = true;
                else
                    okayToGiveBlood = false;
            }
            else if (height == 59)
            {
                if (weight >= 142)
                    okayToGiveBlood = true;
                else
                    okayToGiveBlood = false;
            }
            else
            {
                okayToGiveBlood = true;
            }
        }
        else if (gender == FEMALE_GENDER && age > 22 && weight >= 110)
        {
            okayToGiveBlood = true;
        }
        else if (gender == FEMALE_GENDER)
        {
            okayToGiveBlood = false;
        }
        else
        {
            /* handle all the other genders */
            if (age > 22)
            {
                okayToGiveBlood = true;
            }
            else
            {
                okayToGiveBlood = false;
            }
        }
        if (okayToGiveBlood)
        {
            cout << "Yes, " << name << " you can donate blood." << endl;
        }
        else
        {
            cout << "No, " << name << " you cannot donate blood." << endl;
        }
```

11. [16 points] ExamCardGame is a new card game that uses a subset of the standard deck of playing cards. The deck will include four Aces, four Kings, four Queens, four Jacks, four Nines and four Eights. With an optical sensor that reads cards, a casino is receiving a string for each hand that says, for example, `"a k q j 9 8"`, where each `a` represents an Ace, each `k` represents a King, each `q` represents a Queen, each `j` represents a Jack, each `9` represents a Nine and each `8` represents an Eight. The order of the letters in the string is the order that the cards were dealt to a player. Based on the cards dealt to you, an integer score can be determined where each Ace is worth 5 points, each King is worth 4 points, each Queen is worth 3 points, each Jack is worth 2 points, each Nine is worth 1 point and each Eight is worth nothing.

The owners of casino want to verify the scores of a player's hand. On the next page, write a function to help them; here is its prototype:

```
int cardCounter(string data, int& value, bool& isValid);
```

- **`data` is the string of students arriving and leaving.**
- **`value` has no particular value when the function is called; the function sets it as indicated below.**
- **`isValid` has no particular value when the function is called; the function sets it as indicated below.**

The function should return the number of the cards it processes.

The function sets `isValid` to `true` when `data` is valid, calculating the total score of the cards found in the `data` parameter into the parameter named `value`. The function must set `isValid` to `false` if the `data` string contains any characters other than `a` or `A` , `k` or `K` , `q` or `Q` , `j` or `J` , `9` or `8` or space, setting the value parameter to be 0 in this case regardless of the cards that might be found in the string.

Whether or not `isValid` is set to `true` or `false`, the function should always return the number of cards it processed.

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

```
int v;  bool isValid;

// In this example, the data string has bad characters in it…
assert(cardCounter("&^#$ a q j", v, isValid) == 3 &&  v==0 &&
isValid == false);

// In this example, the data string has the value of ten…
assert(cardCounter("88 aa 88", v, isValid) == 6 && v == 10 &&
isValid == true );

// capitalization doesn’t matter…
assert(cardCounter("88 Aa 88", v, isValid) == 6 && v == 10 &&
isValid == true );
```

11. (continued) Write your `cardCounter` function here. (You do not have to write a main routine or #include directives.)

```
int cardCounter(string data, int& value, bool& isValid)
{







}
```