1. [16 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	F X	G	H
TRUE	TRUE	[✓ TRUE]	[✓ TRUE]	[✓ TRUE]	TRUE	TRUE	TRUE
[✓ FALSE]	[✓ FALSE]	FALSE	FALSE	FALSE	[✓ FALSE]	[✓ FALSE]	[✓ FALSE]
1	2	3	4	5	5	6	7

A. In a C++ program, the body of a `do-while` loop may <u>never</u> be executed.

B. The logical expression: `( x <= y )` ⇒ x<5 || x=5
executes the same as the logical expression: `( x < y && x == y )`

C. The C++ statement: `loop += count--;` loop = loop + count--
has the same effect as the statements:

```
loop = loop + count;
count = count - 1;
```

true & false = false ! false = true

D. Consider the expression: `!(false && (true || false))`
When executed, this expression evaluates to `true`

E. Given the statement: `int x = b * c + f / g - h;`
C++ will perform the multiplication and division before the addition and subtraction.

F. Given the statements:

```
int j( -1 ); j=-1
while (j < 0)    ;
      j = j + 1; j=0
```

When executed, this program will execute an infinite loop and never end.

G. Given the statements:

```
int j( -1 );
do
      j = j + 1; j=0
while( j < 0 );
```

When executed, this program will execute an infinite loop and never end.

H. In C++, when `int` or `double` variables are left uninitialized, C++ will initialize these variables to the value 0

2. [12 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 X	C	D X	X E	F
[✓ TRUE]	TRUE	[✓ TRUE]	TRUE	TRUE	TRUE
FALSE	[✓ 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>        fine
using namespace std;

int main( ) fine
{
     int 123HappyDays( 0 );
     while( int i = 0; i < 123HappyDays; i++ )
            cout << i << endl;
     return( 0 );
}
```

C++ will generate compile errors when trying to compile this code.

C. Given the program:

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

int main( )
{
     char a1( 'a1' ); not a single character, two characters
     cout << a1;
     return( 0 );
}
```

C++ will generate compile errors when trying to compile this code.

? 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 generate compile errors when trying to compile this code.

E. Each of the following output statements will print the value: 15

```
cout << 15 << endl;
cout << 5*3 << endl;
cout << "15" << endl;
char c( '1' );
cout << c;   c = c+4;   cout << c << endl;
```

<= Store int value cuz its a char

F. 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: `UCLA Bruins!`
the program will output: `UCLA Bruins!`

Just UCLA

need getline( )

3. [6 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
[✓ TRUE]	YES	TRUE
FALSE	[✓ NO]	[✓ FALSE]

A. In a C++ program, == is used to test for equality and = is used to assign a value to a variable.

B. You have been asked to print all the integers from 99 down to 0, one value on each line. Consider the following program to complete this task:

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

int main( )
{
    int counter = 99;
    while (counter < 100) => Always true
    {
       cout << counter << endl;
       counter--;
    }
    return( 0 );
}
```

Will this program complete the assigned task successfully? Yes or No.

C. Given the statements:

```
if (i == 12 || j < 20){
     cout << "UCLA ";}
     cout << "Bruin ";
cout << "Bear" << endl;
```

the only possible output that can result from running this code will be either:

```
Bear
```

or Bruin Bear

```
UCLA Bruin Bear
```

4. [6 points]
Write what will be printed by each of the following statements in the space provided.

3·4=12+5=17

a.
```
cout << 3 * 4+5;
```
17

10/4=2 +17=19

b.
```
cout << 17 + 10 / 4;
```
19

c.
```
cout.setf( ios::boolalpha );
int n = 17;
cout << (n == 17);
```
true

true true

d.
```
string s = "17";
cout << s + "10 / 4";
```
concatenated

1710 / 4

e.
```
double d( 12.0000 );
cout << d;
```
12.0000

f.
```
double e( 12.0000 );
cout.setf( ios::showpoint );
cout.precision( 1 );
cout.setf( ios::fixed );
cout << e;
```
12.0

5. [10 points]

1. Write a single statement that declares an array of 8 integers. Name the array `array`.
2. Write a single statement that reads a value from `cin` and stores it in the last element of `array`.
3. Write a loop that sets every element of `array` that has an index that is even to twice the value read from `cin` from Step 2 above (If the input from `cin` were 3, for example, then after this loop has completed, the last element of `array` would have the value 3, and `array [ 0 ]`, `array [ 2 ]`, `array [ 4 ]` and so on... would each have the value 6).

You may assume that:

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

have appeared previously.

1.

```
int array[8];
```

✓

2.

```
cin >> array[7];
```

✓

3.

```
for (int i=0; i<6; i++){
  if (i%2 ==0){
    array[i]=array[7]•2;
  }
}
```

✓

6. [10 points] The following program fragment comes from the Br-Uber assignment that was Programming Project 2.

The Bruin ride-share service named Br-Uber now would like to charge a carbon-footprint offset fee. The fee should be calculated to add 10% to the total cost of each fare. However, no offset fee should apply to rides that are charged just the minimum fare.

Please review the code shown below:

```
cout << "Customer Name: ";
getline( cin, name );
cout << "Miles to be driven: ";
cin  >> mileage;
cout << "Minutes to be driven: ";
cin  >> minutes;
cin.ignore( INT_MAX, '\n' );
cout << "Ride Type: ";
getline( cin, type );

// a bunch of code goes here to verify the data entered...
// and then

if (type == BRUBERX_TYPE)
{
     cost = BOOKING_FEE + mileage * BRUBERX_COST_PER_MILE +
                          minutes * BRUBERX_COST_PER_MINUTE;
     if (cost < BRUBERX_MINIMUM)
     {
          // ensure at least a minimum fare!
          cost = BRUBERX_MINIMUM;
     }
}
else if (type == BRUBERXL_TYPE)
{
     cost = BOOKING_FEE + mileage * BRUBERXL_COST_PER_MILE +
                          minutes * BRUBERXL_COST_PER_MINUTE;
     if (cost < BRUBERXL_MINIMUM)
     {
          // ensure at least a minimum fare!
          cost = BRUBERXL_MINIMUM;
     }
}

// add the carbon-footprint offset fee
cost = cost + cost * 1 / 10;

cout.precision( 2 );
cout.setf( ios::fixed );
cout.setf( ios::showpoint );
cout << "The fare for " << name << " is $" << cost << endl;
```

```
else{ cost= cost + (.1) cost; }

else{ cost = cost + (.1) cost; }
```

6. (continued)
Unfortunately, the program does not work as intended. What small changes to the program will fix its problems?
Either clearly indicate the changes that need to be made below or rewrite a portion of the code below.

```
cout << "Customer Name: ";
getline( cin, name );
cout << "Miles to be driven: ";
cin  >> mileage;
cout << "Minutes to be driven: ";
cin  >> minutes;
cin.ignore( INT_MAX, '\n' );
cout << "Ride Type: ";
getline( cin, type );

// a bunch of code goes here to verify the data entered...
// and then

if (type == BRUBERX_TYPE)
{
     cost = BOOKING_FEE + mileage * BRUBERX_COST_PER_MILE +
                          minutes * BRUBERX_COST_PER_MINUTE;
     if (cost < BRUBERX_MINIMUM)
     {
          // ensure at least a minimum fare!
          cost = BRUBERX_MINIMUM;
     }
}
else if (type == BRUBERXL_TYPE)
{
     cost = BOOKING_FEE + mileage * BRUBERXL_COST_PER_MILE +
                          minutes * BRUBERXL_COST_PER_MINUTE;
     if (cost < BRUBERXL_MINIMUM)
     {
          // ensure at least a minimum fare!
          cost = BRUBERXL_MINIMUM;
     }
}

~~// add the carbon-footprint offset fee~~
~~cost = cost + cost * 1 / 10;~~

cout.precision( 2 );
cout.setf( ios::fixed );
cout.setf( ios::showpoint );
cout << "The fare for " << name << " is $" << cost << endl;
```

take this out

~~ADD~~: else { cost = cost + (.1)*cost; }

ADD: else { Cost = cost + (.1)*cost; }

7. [5 points] Convert this switch statement to code that produces <u>exactly the same output</u> 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 'U':
     cout << "University " << endl;
     break;
  case 'C':
     cout << "California " << endl;
  case 'L':
  case 'A':
     cout << "Los Angeles " << endl;
     break;
  default:
     cout << "Bruin!" << endl;
     break;
}
```

```
if (letter == 'U'){
  cout << "University " <<endl;
}
else if (letter=="C"){
  cout << "California "<<endl;
}
else if (letter== 'L' || letter=='A'){
  cout << "Los Angeles " <<endl;
}
else {
  cout << "Bruin!" << endl;
}
```

-1

? 8. [5 points] Which of these statements is true about the following program? (Circle a, b, or c; if you circle b or c, briefly state the problem)

a. It is a well-formed C++ program with well-defined behavior if the user enters `COLLEGE STUDENT`.

3 4 5 6 7 8 9 10 11 12 13 14 15

[✓ b.] It has no error that would prevent compilation, but when executed, if the user enters `COLLEGE  STUDENT`, it has undefined behavior (e.g., it might crash or produce strange results). Briefly state what the problem is:

Text[0], text[12] and text[4] are characters, on different system characters have different corresponding int values, such as ASCII. The compiler will force the char to its corresponding int value for arithmetic, but depending on the encoding it might not be the value expected.

c. It has at least one error that will prevent compilation from succeeding. Briefly state what the problem is:

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

int main( )fine
{
    cout << "Enter a kind of person:  ";fine
    string text;fine
    getline(cin, text);fine
                          → size type vs int comparison
    if ( text.size() >= 12   &&
         text[0] / ( text[12] - text[4] ) > 10 )
char typed forced to int  cout << "Done" << endl;fine ✗   Division by 0

    return( 0 );fine
}
```

9. [20 points] The owners of a dog kennel are trying to keep track of dog drop-offs and dog pickups by their customers. With a sensor as customers enter and leave, they are receiving a string each day that says, for example, `"dd  Dd  DD"`, where each `d` represents that a dog has been dropped off by its owner, and each `D` means that a dog has been picked up by its owner and spaces mean nothing has changed. The order of the letters in the string is the order that dogs have come and gone from the kennel. The string `"dd  Dd  DD"` means that first 2 dogs were dropped off, then eventually 1 got picked up and 1 more dog got dropped off, and then eventually 2 dogs got picked up.

The owners of the dog kennel would like to know that their kennel is operating well within its approved size. On the next page, write a function to help them; here is its prototype:

```
bool dogKennel(string data, int kennelSize, int& maximum);
```

`data` is the string of dog drop offs and pickups.
`kennelSize` is the maximum number of dogs who can be safely accommodated in the kennel simultaneously.
`maximum` has no particular value when the function is called; the function sets it as indicated below.

The function returns `true` if the maximum number of dogs in the kennel simultaneously never exceeded the `kennelsize`, or `false` if that maximum did exceed the `kennelsize` at some point. Regardless of the return value, the function must set `maximum` to the maximum number of dogs who were in the dog kennel at any one time.

Notwithstanding the above paragraph, the function must also return `false` if the `data` string contains any characters other than `d` or `D` or space, if `kennelSize` is negative, or if during the analysis of the data, it appears that at some point there were a negative number of dogs in the kennel.

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

```
int max;
assert(dogKennel("dd   DdD D", 10, max) &&  max == 2);

// In this example, the maximum number of dogs in the kennel at
// once was 8 which exceeded the kennelSize of 6
assert(!dogKennel("d D dd dd dd dd D", 6, max) &&  max == 8);

// In this example, 2 dogs were dropped off and then 10 dogs
// were picked up, leading to a negative number of dogs in the
// kennel at one point!
assert(!dogKennel("dd  DDDDD DDDDD", 20, max) && max == 2);
```

c

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

```
bool dogKennel(string data, int kennelSize, int& maximum)
{  max=0;
  if (kennelSize <0){ //checking for negative kennel size
    return false;

  }
  for(size_t i=0; i<data.size(); i++){      //checking to make sure well formed data input
    if (data[i]!= ' ' && data[i]!='d' && data[i]!='D') {
      return false;
    }

  } int numdog=0;
  for(size_t i=0; i<data.size(); i++){
      if (data[i]=='d'){
      numdog++;
      }
      if (data[i]=='D'){
      numdog--;
      }
      if(data[i]==' '){
       if(numdog >kennelSize){
         return false;
       }
       if(maximum <numdog){
        maximum=numdog;
       }
       if( numdog <0){
        return false;
       }
  } }  }
  if (maximum <kennelSize){
    return true;
  }
```

closing for loop

closing for loop

closing if

-5

ALL PATHS
NEED TO
RETURN
BOOL

10. [10 points] What is the last digit of your UCLA student ID number? 6
What output is produced by the following program if its input is the last digit of your UCLA student ID number? Please use the lines below to show what you think will be printed by the program.

6-292017 50-7

50-7

7-3

50-6

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

int bruin(int one, int& two);
void bear(int& three);

int main()
{
    int data;
    cin >> data; // enter the last digit of your student id
    int today = 292017;
  1. today = bruin(today, data);  today is set equal to 50
  2. bear(today); now data = 7 and today = 50
  3. bruin(today,data); now data = 7 and today = 3
}

                 3       7
int bruin(int one, int& two)
{
              7 - 3
    cout << two << "-" << one << endl;
    two--; 6 = data
    one = 50;
    cout << one << "-" << two << endl;
    return( one );
}

                  50
void bear(int& three)
{
    if (three > 100) no, 100>50
        three = 7;
    else if (three > 7) yes, 50>7
        three = 3;
    else if (three > 4)
        three = 1;
}
        three = today = 3
```