CSI 201
practice, if-else, else-if nests!,

1. Goals for today: Practice, Explore, Practice!

2. Recall the syntax of an if-else statement

```
double a = 0, b = 0; //declares and initializes
cin >> a >> b;
if(a > b) {  //is a greater than b? if so then do:
  cout << "a is greater than b" << endl;
}
else { //well a must not have been greater than b, so ...
  cout << "b is greater than (or equal to) a" << endl;
}
```

3. In this worksheet, we'll first explore a few if statements, then we'll move on to a few if-else statements. Lastly, we'll try a few advanced concepts like nested if statements and else if statements.

if statements (else options are in blue and italics).

1. Create a variable to represent the number of whole seconds since January 1st, 1970 (also called UNIX epoch time). Then, read in a value for this variable from the user. When the value the user gives is less than 1517374800 then output "that was in the past". *Otherwise output "today or in the future".*

2. Create a variable to represent the user's height (in inches) to be used in a carnival ride. Read in the value from the user. If the user is not at least 44 inches (the height required for Space Mountain at Walt Disney World), then report that the user cannot go on the ride and quit the program. At the end of your program, output something exciting to indicate the user's fun on this roller coaster.

3. Create a variable to represent a number of people. The provost of a school might determine that a class cannot be run unless it has 6 people in it. Read in your variable from the user. Write a conditional to determine if there are enough people to run the class and output something positive if there are enough people! *Otherwise output there are not enough people.*

4. Create a variable to represent a user's percentage on a test. If that percentage is higher than 60, then report something exciting to indicate the user has passed the test! *Otherwise output something encouraging for the user.*

5. Create a variable for the user that represents the user's best guess at the number the computer will choose. Use the starter code below. It will get a random number from 1-10 into the variable random_number. Then, output "YES!" if the user input guess is equal to the variable random_number. (To test equality, we use == and not single =). *Otherwise, output NOT EQUAL!*

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

int main() {
  srand(time(0));
  int random_number = rand()%10 + 1;
}
```

6. Read in a double, called squareField from the user. If squareField is positive, then output the length of one side of the field by using: `sqrt(squareField)`. If you end up coding this in zybooks, you'll need to `#include<cmath>` at the top of your code. *Otherwise, output, negative area is nonsense!*

7. Read in hitpoints from the user and then read in damage from the user. If damage is larger than hitpoints, then output "YOU ARE DEAD!" *Otherwise, modify hitpoints to be its value minus the damage. Lastly, output the new hitpoints value to the user.*

nested if statements

As an example:

```
int age = 23;
cin >> age;
if(age >= 18) { //is the user >= 18?
  cout << "You're an adult, technically" << endl;
  if(age >= 21) { //age is >= 18 AND >= 21
    cout << "You're also technically allowed to drink now!" << endl;
    if(age >= 25) { //age is >= 18 AND >= 21 AND >= 25
      cout << "Your brain isn't making many new connections!" << endl;
    }
  }
}
```

1. Create a variable to represent the number of whole seconds since January 1st, 1970. Then, read in a value for this variable from the user. When the value the user gives is more than 1517374800 but not more than 1517461200 then output "That's TODAY!" There are a few ways to solve this (one of which involves nesting if statements).

2. Read in two integers from the user called a and b. if a is more than 0 and b is more than 0, then output "BOTH POSITIVE" Lastly, if the result of `sqrt(a*b)` is equal to a, then output "perfect square!"

3. Get a random number from 1-20 from the user. Start with the starter code from #5 above and change the 10 to a 20 to get a number from 1-20. If the value on the dice is more than 13 then output "YOU HIT". If the value is also more than 17 then output "YOU GOT A SPECIAL EFFECT". Lastly, if the value is also a 20 then also output "YOU CRITICALLY HIT!

else if statements

As an example:

```
char a; //you can read in just a single character!
cin >> a;
if(a == 'y') { //if a is a y
  cout << "user typed lower y" << endl;
}
else if(a == 'n') { //if a wasn't a y but is it an n?
  cout << "user typed lower n" << endl;
}
else { // a wasn't a y AND a wasn't a n
  cout << "user typed neither a lower n nor a lower y" << endl;
}
```

1. Start with your code from problem #5 (with srand in it) from above. But now, determine if the number is less than (and output you guessed low), greater than (and output you guessed high) or equal to (and output YES!).

2. Read in an integer percentage from the user. Output a A, B, C, D or F based on the grade the user enters.

3. Get a random number from 1-20 from the user. Use the starter code in #5 above. If the random number is a 1, output critical miss! Otherwise, if the random number is below 13 but not a 1, then output miss! Otherwise, if the number is below 19 (but more than 12), output hit. Lastly, if the number is 20 output CRITICAL HIT!