`do`/`while` and Nested Loops

Nested Loops

- A loop within a loop
- Can repeat multiple things within a loop
- Example:
  - Read in 10 grades for each of 20 students
  - How would we write loops to do this?

What is the Output?

```
for(int i = 0; i < 3; i++)
{
  for(int j = 0; j < 2; j++)
  {
    // how many times will this run?
    cout << "*";
  }
  // how many times will this run?
  cout << endl;
}
```

What is the Output?

```
for(int i = 3; i > 0; i--)
{
  for(int j = 0; j < i; j++)
  {
    cout << "*";
  }
  cout << endl;
}
```


Practice

- Write C++ statements that will read in an integer from the user (n) and produce:

```
1
22
333
4444
....
nnnnnnnnn
```


`do/while` Repetition Structure

- So far, we have looked at
  - `while`
  - `for`
  - both of these are pretest loops
- `do/while` is another repetition structure
- Post-test: test happens at the end of the loop

do/while Loops

```
do
{
  cout << "Enter a year:" << endl;
  cin >> year;
} while (year < 0); // TEST!
// The body of the loops happens
// before the test
```


When to use do while?

- When loop must execute at least once
- Perfect for data validation!
- Post-tested loop

```
do
{
  statements;
} while ( expression );
```


What's the output?

```
int m = 10;
do
{
  cout << m << endl;
  m = m - 3;
} while (m > 0);
```

Example

- Write C++ statements that will read in integers from the user until the user inputs an integer between 5 and 10.

Rewrite as a do/while loop

```
int num = 0;
// what is the output?
while (num <= 6)
{
  cout << num << endl;
  num += 2;
}
```


Rewrite as a do/while Loop

```
// what is the output?
for (n = 3; n > 0; n--)
{
  cout << n << " squared is"
       << n * n << endl;
}
```

What is the Output?

```
int counter = 1;
do
{
  cout << counter << " ";
} while( ++counter <= 10 );
```

```
int counter = 1;
do
{
  cout << counter << " ";
} while( counter++ <= 10 );
```


What is the Purpose?

```
char ch;
do
{
  cout << "Enter a valid code (h,c,i): ";
  cin >> ch;
} while ((ch != 'h') && (ch != 'c') && (ch != 'i'));
```