Loops

- while
- do-while
Miscellaneous

- Project 3: Must submit test cases
- Submit as text files (.txt) only
**Miscellaneous**

- Project 3: Must submit test cases
- Submit as *text* files (.txt) only

- **Start working on your projects early**
Miscellaneous

- Project 3: Must submit test cases
- Submit as *text* files (.txt) only

- Start working on your projects early
  - `i++`
  - `j += 2`
i++, j+=2

i++;  
i = i + 1;

i--;  
i = i - 1;

i+=2;  
i = i + 2;

i-=2;  
i = i - 2;
Miscellaneous

- Project 3: Must submit test cases
- Submit as *text* files (.txt) only

- Start working on your projects early

- `i++`
- `j+=2`

- *unsigned char, unsigned int*
- *literal*
unsigned char, unsigned int
Write an if/else statement that adds 1 to the variable minors if the variable age is less than 18, adds 1 to the variable adults if age is 18 through 64, and adds 1 to the variable seniors if age is older than 64.

```java
if ( age < 18 )  {
    minors = minors + 1;
} else if ( 18 <= age <= 64 ) {
    adults = adults + 1;
} else if ( age > 64 ) {
    seniors = seniors +1;
}
```
CodeLab

Write an if/else statement that adds 1 to the variable minors if the variable age is less than 18, adds 1 to the variable adults if age is 18 through 64, and adds 1 to the variable seniors if age is older than 64.

```java
if ( age < 18 ) {
    minors = minors + 1;
} else if ( 18 <= age <= 64 ) {
    adults = adults + 1;
} else if ( age > 64 ) {
    seniors = seniors +1;
}

BUT WHAT AM I DOING WRONG ????
```
Write an if/else statement that adds 1 to the variable minors if the variable age is less than 18, adds 1 to the variable adults if age is 18 through 64, and adds 1 to the variable seniors if age is older than 64.

```java
if ( age < 18 ) {
    minors = minors + 1;
} else if ( 18 <= age <= 64 ) {
    adults = adults + 1;
} else if ( age > 64 ) {
    seniors = seniors + 1;
}
```

BUT WHAT AM I DOING WRONG ????
Write an if/else statement that adds 1 to the variable minors if the variable age is less than 18, adds 1 to the variable adults if age is 18 through 64, and adds 1 to the variable seniors if age is older than 64.

```java
if ( age < 18 ) {
    minors = minors + 1;
} else if ( 18 <= age && age <= 64 ) {
    adults = adults + 1;
} else if ( age > 64 ) {
    seniors = seniors +1;
}
```

BUT WHAT AM I DOING WRONG ?????
Write an if/else statement that adds 1 to the variable minors if the variable age is less than 18, adds 1 to the variable adults if age is 18 through 64, and adds 1 to the variable seniors if age is older than 64.

```java
if ( age < 18 ) {
    minors = minors + 1;
} else if (( 18 <= age ) && ( age <= 64 )) {
    adults = adults + 1;
} else if ( age > 64 ) {
    seniors = seniors +1;
}
```

CAN I MAKE THIS MORE EFFICIENT ????
Write an if/else statement that adds 1 to the variable minors if the variable age is less than 18, adds 1 to the variable adults if age is 18 through 64, and adds 1 to the variable seniors if age is older than 64.

```java
if ( age < 18 ) {
    minors = minors + 1;
} else if ( 18 <= age ) && ( age <= 64 ) {
    adults = adults + 1;
} else if ( age > 64 ) {
    seniors = seniors +1;
}
```

CAN I MAKE THIS MORE EFFICIENT ????
Write an if/else statement that adds 1 to the variable minors if the variable age is less than 18, adds 1 to the variable adults if age is 18 through 64, and adds 1 to the variable seniors if age is older than 64.

```java
if ( age < 18 ) {
    minors = minors + 1;
} else if ( age <= 64 ) {
    adults = adults + 1;
} else if ( age > 64 ) {
    seniors = seniors + 1;
}
```

CAN I MAKE THIS MORE EFFICIENT ????
CodeLab

Write an if/else statement that adds 1 to the variable minors if the variable age is less than 18, adds 1 to the variable adults if age is 18 through 64, and adds 1 to the variable seniors if age is older than 64.

```java
if ( age < 18 ) {
    minors = minors + 1;
} else if ( age <= 64 ) {
    adults = adults + 1;
} else if ( age > 64 ) {
    seniors = seniors + 1;
}
```

CAN I MAKE THIS MORE EFFICIENT ????
Write an if/else statement that adds 1 to the variable minors if the variable age is less than 18 , adds 1 to the variable adults if age is 18 through 64 , and adds 1 to the variable seniors if age is older than 64.

```java
if ( age < 18 ) {
    minors = minors + 1;
} else if ( age <= 64 ) {
    adults = adults + 1;
} else if ( age > 64 ) {
    seniors = seniors +1;
}
```

**CAN I MAKE THIS MORE EFFICIENT ?????**
Write an if/else statement that adds 1 to the variable minors if the variable age is less than 18, adds 1 to the variable adults if age is 18 through 64, and adds 1 to the variable seniors if age is older than 64.

```java
if ( age < 18 ) {
    minors = minors + 1;
} else if ( age <= 64 ) {
    adults = adults + 1;
} else {
    seniors = seniors + 1;
}
```
while loop

while ( <expression> )
<statement>

evaluate expression

evaluate expression

ture

execute loop body

false

EXIT

true

EXIT
Example 1

int number = 1;

while ( number <= 5 )
{
    cout << "Hello";
    number++;
}

cout << "\nThat’s all folks!";
Example 1

int number = 1;

while ( number <= 5 )
{
    cout << “Hello”;  
    number++;  
}

cout << “\nThat’s all folks!”;  

// Prints hello how many times?
Example 1

```cpp
int number = 1; // initialize loop control variable

while ( number <= 5 ) { // loop while conditional ‘true’
    cout << “Hello”; // update loop control variable
    number++;
}

cout << “\nThat’s all folks!”;

// Prints hello how many times?
```
Example 1

```cpp
int number = 1;  // initialization

while ( number <= 5 )  // condition
{
    cout << “Hello”;
    number++;  // update
}
```
Example 2

```cpp
int number = 6;

while ( number <= 5 )
{
    cout << "Hello";
    number++;
}

// Prints hello how many times?
```
Example 3

int number = 1;

while ( number <= 5 )
{
    cout << "Hello";
    //number++;
}

// Prints hello how many times
Example 4

```cpp
int number = 1;

while ( number <= 50 )
{
    cout << "Hello";
    number++;
}

// Prints hello how many times
```
Example 5

```cpp
int number = 1;

while ( number <= 50 ) {
    cout << "Hello";
    number++;
}

// Prints hello how many times
```
Great for Checking Validity of Input

```cpp
int getData()
{
    int number;
    cout << "Enter a number in range 1-100: ";
    cin >> number;
    while ( number < 1 || number > 100 )
    {
        cout << "ERROR: Enter a value in range 1-100: ";
        cin >> number;
    }
    return number;
}
```
int getData()
{
    int number;
    cout << "Enter a number in range 1-100: ";
    cin >> number;

    while ( number < 1 || number > 100 )
    {
        cout << "ERROR: Enter a value in range 1-100: ";
        cin >> number;
    }

    return number;
}
int getData() {
    int number;
    cout << "Enter a number in range 1-100: ";
    cin >> number;

    while ( number < 1 || number > 100 ) // condition
    {
        cout << "ERROR: Enter a value in range 1-100: ";
        cin >> number;
    }

    return number;
}
int getData()
{
    int number;
    cout << "Enter a number in range 1-100: ";
    cin >> number;                    // initialization

    while ( number < 1 || number > 100 )
    {
        cout << "ERROR:  Enter a value in range 1-100: ";
        cin >> number;
    }

    return number;
}
int getData()
{
    int number;
    cout << "Enter a number in range 1-100: ";
    cin >> number;

    while ( number < 1 || number > 100 )
    {
        cout << "ERROR: Enter a value in range 1-100: ";
        cin >> number;             // update
    }

    return number;
}
Great for Checking Validity of Input

```cpp
int getData()
{
    int number;
    cout << "Enter a number in range 1-100: ";
    cin >> number;

    while ( number < 1 || number > 100 )
    {
        cout << "ERROR: Enter a value in range 1-100: ";
        cin >> number;
    }

    return number;
}
```
Do While – A Change in Order

```plaintext
do
    <statement>
while (<true>);
```

**Do – While** loops always execute the loop body at least once before checking the condition.
Check Validity Using do-while

```cpp
int getData()
{
    int number;

    do {
        cout << "Enter a number in range 1-100: ";
        cin >> number;
    } while ( number < 1 || number > 100 )

    return number;
}
```
int getData() {
    int number;

    do {
        cout << "Enter a number in range 1-100: ";
        cin >> number;
    } while ( number < 1 || number > 100 )

    return number;
}
Check Validity Using do-while

```cpp
int getData()
{
    int number;

    do {
        cout << "Enter a number in range 1-100: ";
        cin >> number;
    } while ( number < 1 || number > 100 ) // condition

    return number;
}
```
int getData()
{
    int number;

    do {
        cout << "Enter a number in range 1-100: ";
        cin >> number; // update
    } while ( number < 1 || number > 100 )

    return number;
}
Check Validity Using do-while

```cpp
int getData()
{
    int number;

    do {
        cout << "Enter a number in range 1-100: ";
        cin >> number;
    } while ( number < 1 || number > 100 )

    return number;
}
```
Note:

• a do-while can ALWAYS be written as a while
Exam on Wednesday

- Time, location on CTools

- Alternate exam
  - Email 183help@umich.edu by the end of **Friday**
  - Explain why you want to take the alternate exam

- Exam review session: Time, location on CTools

- Practice last year’s exams
- Read lecture notes
Exam on Wednesday

- Part 1: Multiple Choice Questions (100 points)
- Part 2: Analyze / Write Small Programs (100 points)
**Sum of first n positive integers**

```c
int sum(int n) {
    int sum = 0;
    int i = 1;
    while (i <= n) {
        sum = sum + i;
        i = i + 1;
    }
    return sum;
}
```
Sum of first n positive integers

// Requires:
// Effects:

int sum(int n)
{
}

}
Sum of first n positive integers

// Requires: n > 0
// Effects: Returns the sum of the first n integers
// That is, returns 1 + 2 + ... + n

int sum(int n)
{
    int sum = 0;
    int i = 1;
    while (i <= n)
    {
        sum = sum + i;
        i = i + 1;
    }
    return sum;
}
Sum of first n positive integers

// Requires: n > 0
// Effects: Returns the sum of the first n integers
// That is, returns 1 + 2 + ... + n
int sum(int n)
{
}

}
// Requires: n > 0
// Effects: Returns the sum of the first n integers
// That is, returns 1 + 2 + ... + n
int sum(int n)
{

    while ( )
    {

    }

}


Sum of first n positive integers

// Requires: n > 0
// Effects: Returns the sum of the first n integers
  // That is, returns 1 + 2 + ... + n
int sum(int n)
{

  int i = 1;
  while ( i <= n )
  {

    i = i + 1;

  }

}
Sum of first n positive integers

// Requires: n > 0
// Effects: Returns the sum of the first n integers
//          That is, returns 1 + 2 + ... + n
int sum(int n)
{
    int s = 0;
    int i = 1;
    while ( i <= n )
    {
        s = s + i;
        i = i + 1;
    }
    return s;
}
Sum of first n positive integers

// Requires: n > 0
// Effects: Returns the sum of the first n integers
//           That is, returns 1 + 2 + ... + n
int sum(int n)
{
    int s = 0;
    int i = 1;
    while ( i <= n )
    {
        s = s + i;
        i = i + 1;
    }
    return s;
}
Sum of first n positive integers

// Requires: n > 0
// Effects: Returns the sum of the first n integers
// That is, returns 1 + 2 + ... + n
int sum(int n) {
    int s = 0;
    int i = 1;
    while (i <= n) {
        s = s + i;
        i = i + 1;
    }
    return s;
}