Selection Statements

- **Decision behavior**
- **Select statements to execute**
  - One statement (or none)
    - Simple or compound statement
  - One of two alternatives
  - One of many alternatives

Simple `if` Statement

```
float taxToPay (float salary)
{
  float tax = 0.0;

  if (salary >= 20000)
        tax = salary * 0.03;

  return tax;
}
```

Condition

Statement

Another `if` Statement

```
small = first;
large = second;

if (first > second)
{
  small = second;
  large = first;
}
```

Condition

Statement

if/else Statement

```
if (first > second)
{
  small = second;
  large = first;
}
else
{
  small = first;
  large = second;
}
```

Condition

Statement-T

Statement-F

if Statement Options

```
small = first;
large = second;
```

```
if (first>second)
{
  small = second;
  large = first;
}
```

```
if (first>second)
{
  small = second;
  large = first;
}
else
{
  small = first;
  large = second;
}
```

Nested if Statements (1)

Nesting structure clear?

Indentation hard to follow if many levels

```
if (avg >= 90.0)
  grade = 'A';
else
  if (avg >= 80.0)
    grade = 'B';
  else
    if (avg >= 70.0)
      grade = 'C';
    else
      if (avg >= 60.0)
        grade = 'D';
      else
        grade = 'F';
```

Nested `if` Statements (2)

Same meaning as previous

Mutually exclusive options

```
if (avg >= 90.0)
  grade = 'A';
else if (avg >= 80.0)
  grade = 'B';
else if (avg >= 70.0)
  grade = 'C';
else if (avg >= 60.0)
  grade = 'D';
else
  grade = 'F';
```

What About This?

Does this do what we want?

`else` needed for mutual exclusion

```
if (avg >= 90.0)
  grade = 'A';
if (avg >= 80.0)
  grade = 'B';
if (avg >= 70.0)
  grade = 'C';
if (avg >= 60.0)
  grade = 'D';
else
  grade = 'F';
```

`if/else` Association

Are these forms different?

```
if (gender == 'M')
  if (age < 21)
    prem = 800.0;
else
  prem = 400.0;
```

`else` matches nearest `if` in same block scope

```
if (gender == 'M')
  if (age < 21)
    prem = 800.0;
  else
    prem = 400.0;
```

Boolean Expressions

- **True/false values**
  - Boolean algebra - George Boole
- **C++ data type and literals**
  - `bool`, `true`, `false`
- **Operators**
  - Relational/equality
  - Logical (Boolean)

Relational/Equality Operators

<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to
!=	Not equal to

Logical Operators

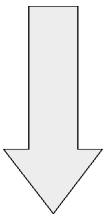
||	Logical OR
&&	Logical AND
!	Logical NOT

```
temp > 90.0||humidity == 90

code != 10 && count < 4
```

Operator Precedence

```
(...)
  !
* / %
 + -
< > <= >=
== !=
  &&
  ||
```



Short-Circuit Evaluation

```
if (a != 0 && (b/a > 3))
  a = 4;
```

```
if ((b/a > 3) && a != 0)
  a = 4;
```

What if a is zero?

If answer can be determined from first operand, second is not evaluated.

Boolean Types and Integers

- **Real Boolean type**
  - In ANSI C++, type `bool`
- **Carry over from C**
  - Integers as Booleans
    - Non-zero = TRUE (1 ??)
    - Zero = FALSE
  - `BOOL` in MS Windows

Assignment & Equality

- **Assignment (=)**
- **Equality (==)**
- **Watch out for confusion!**

```
int a = 7;
if (a = 2)
{
  cout << "Yes" << endl;
}
```