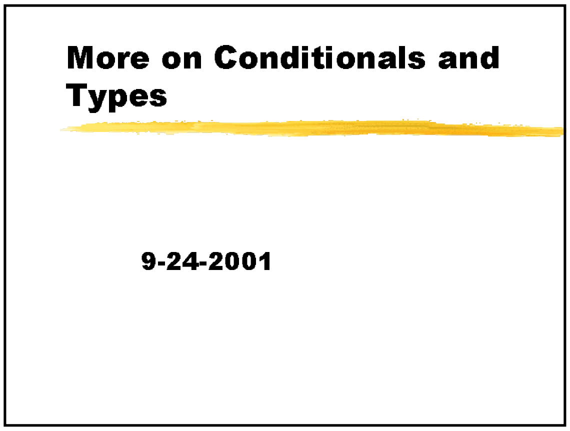
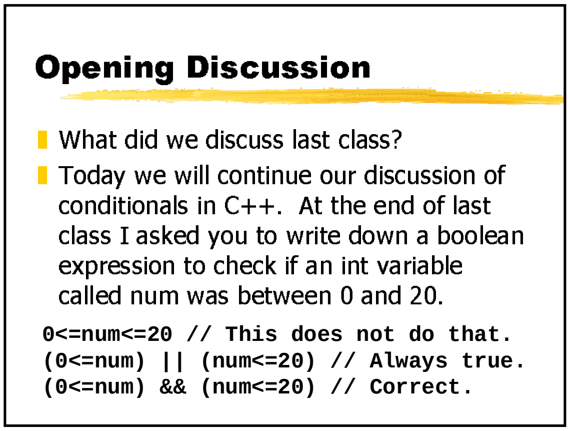


More on Conditionals and Types

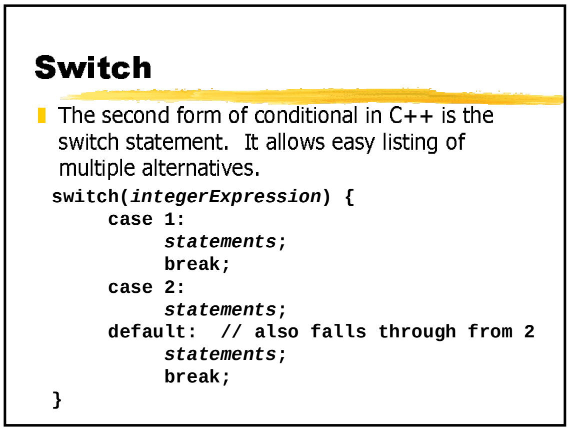
9-24-2001



Opening Discussion

- What did we discuss last class?
- Today we will continue our discussion of conditionals in C++. At the end of last class I asked you to write down a boolean expression to check if an int variable called num was between 0 and 20.

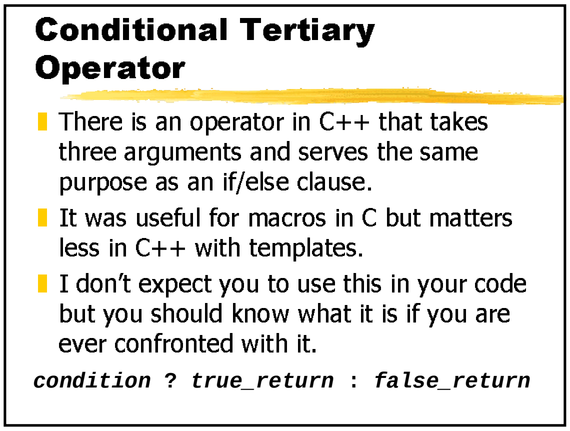
```
0<=num<=20 // This does not do that.
(0<=num) || (num<=20) // Always true.
(0<=num) && (num<=20) // Correct.
```



Switch

- The second form of conditional in C++ is the switch statement. It allows easy listing of multiple alternatives.

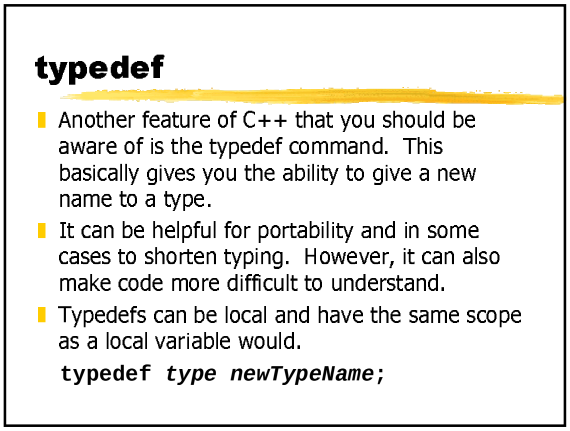
```
switch(integerExpression) {
     case 1:
          statements;
          break;
     case 2:
          statements;
     default:  // also falls through from 2
          statements;
          break;
}
```



Conditional Tertiary Operator

- There is an operator in C++ that takes three arguments and serves the same purpose as an if/else clause.
- It was useful for macros in C but matters less in C++ with templates.
- I don't expect you to use this in your code but you should know what it is if you are ever confronted with it.

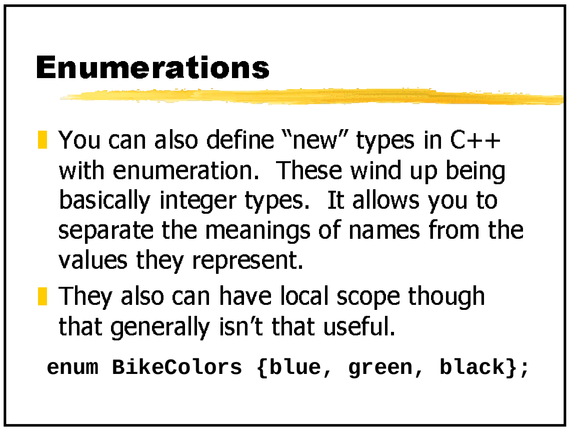
```
condition ? true_return : false_return
```



typedef

- Another feature of C++ that you should be aware of is the typedef command. This basically gives you the ability to give a new name to a type.
- It can be helpful for portability and in some cases to shorten typing. However, it can also make code more difficult to understand.
- Typedefs can be local and have the same scope as a local variable would.

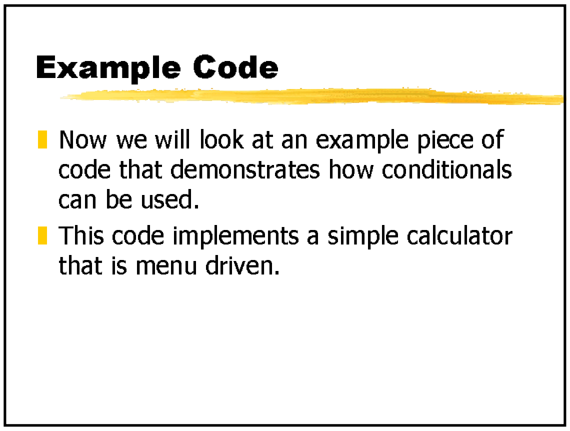
```
typedef type newTypeName;
```



Enumerations

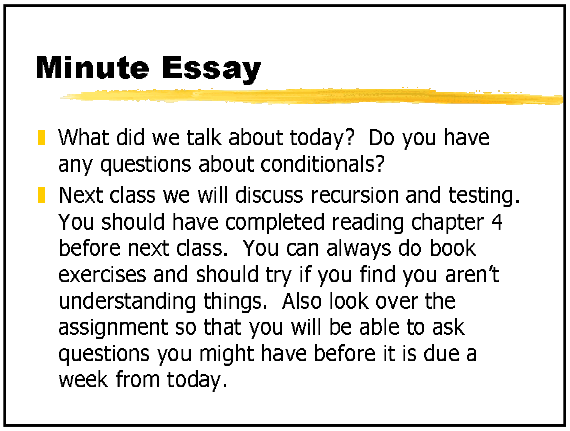
- You can also define "new" types in C++ with enumeration. These wind up being basically integer types. It allows you to separate the meanings of names from the values they represent.
- They also can have local scope though that generally isn't that useful.

```
enum BikeColors {blue, green, black};
```



Example Code

- Now we will look at an example piece of code that demonstrates how conditionals can be used.
- This code implements a simple calculator that is menu driven.



Minute Essay

- What did we talk about today? Do you have any questions about conditionals?
- Next class we will discuss recursion and testing. You should have completed reading chapter 4 before next class. You can always do book exercises and should try if you find you aren't understanding things. Also look over the assignment so that you will be able to ask questions you might have before it is due a week from today.