CS 161
Intro to CS I

User Input and Conditional Statements
We can use a switch...

```java
switch( <expression> ) {
    case <const-expr>:
        <statement>;
        ...
    ...
    case <const-expr>:
        <statement>;
        ...
    ...
    default:
        <statement>;
        ...
}
```
C++ Switch Example

```cpp
switch( x ) {
    case 0:
        std::cout <<“X is zero\n”;
        break;
    case 1:
        std::cout <<“X is one\n”;
        break;
    case 2:
        std::cout <<“X is two\n”;
        break;
    default:
        std::cout <<“You have entered an invalid number!!!\n”;
}
```
C++ Switch Example

```cpp
switch( x ) {
    case 0:
    case 1:
        std::cout << "X is zero or one\n";
        break;
    case 2:
        std::cout << "X is two\n";
        break;
    default:
        std::cout << "You have entered an invalid number!!!\n";
}
```
Demo...
New Operators for Expressions

• What if you have the statement \texttt{var = var + 1;}
  \texttt{var += 1; /*Add operand on right to var*/}
  \texttt{var++; /*Increment var by one*/}

• What if you have the statement \texttt{var = var - 5;}
  OR \texttt{var = var + var;}
  \texttt{var -= 5;}
  \texttt{var += var;}

• Pre vs. Post increment: \texttt{++var} vs. \texttt{var++}
Demo...
Multiple Decisions

• What if I want to make these same decisions for the whole year?
  If it is sunny today
    then I’ll go to the beach
    if it is windy at the beach
      then I’ll fly a kite
    else if it is not windy at the beach
      then I’ll walk on the shore
  Else if it is raining today
    then I’ll stay inside and read a book
  Else if it is snowing
    then I’ll go to the mountains to ski

• Repeat the process for 365 days
How do we do this?

• Repetition: for loops
  – Semantics
    • Repeat for a specific # of iterations w/ starting point, ending point, and an increment
  – Syntax
    for(x=1; x <= 365; x++) {
      <statement>;
      <statement>;
      ...
    }
The for Loop

for(x=1; x <= 365; x++) {
    <statement>;
    <statement>;
    ...
}

Starting point:
Initialization
The for Loop

for(x=1; x <= 365; x++) {
    <statement>;
    <statement>;
    ...  
}

Ending point: Continuation Test
The for Loop

for(x=1; x <= 365; x++) {
  <statement>;
  <statement>;
  ...
}

• What do you notice about order?
The for Loop

```java
for(x=1; x <= 365; x++) {
    <statement>;
    <statement>;
    ...

}  
```

- Same as `x = x+1`
- What about `x = x + 2`?
The for Loop

for(x=1; x <= 365; x++) {
    <statement>;
    <statement>;
    ...
}

• What do you notice about order?
The for Loop

```java
for(x=1; x <= 365; x++) {
    <statement>;
    <statement>;
    ...
}
```

Test is False: Execution after loop
The for Loop Examples

```c
for(x=-100; x <= 100; x++)
    cout << "hello world\n";
for(x=2+2; x <= 17*3; x++)
    cout << "hello world\n";
for(x=0; x <= 100; x++)
    cout << "hello world\n";
for(x=0; x < 100; x++)
    cout << "hello world\n";
for(x=-100; x <= -1; x++)
    cout << "hello world\n";
```
The for Loop Examples

for(x=1; x <= 1; x++) {
    cout << “hello world\n”;    
}  

for(x=1; x < 1; x++) {
    cout << “hello world\n”;   
}  

• Why is it better to use curly braces?
The for Loop Pattern

for(<variable> = n; <variable> <= p; <variable>++) {
    <statement>;
    ...
}

for(<variable> = n; <variable> >= p; <variable>--) {
    <statement>;
    ...
}
Nested for Loops

```cpp
for(x = 0; x < 10; x++) {
    for(y = 0; y < 10; y++) {
        cout << "hello world\n";
    }
}
```

• How many times is Hello World printed?
Reuse Variables

for(x = 0; x < 10; x++) {
    cout << "The value of x is: " << x << endl;
}
for(x = 0; x < 10; x++) {
    cout << "The value of x is: " << x << endl;
}
Variables with same name

int x;
for(x = 0; x < 10; x++) {
    for(x = 0; x < 10; x++) {
        cout << "The value of x is: " << x << endl;
    }
}

• What is the output from this nested loop?
Infinite Loops

```cpp
int x;
for(x = 0; x < 10; x++) {
    for(x = 0; x < 5; x++) {
        cout << "The value of x is: " << x << endl;
    }
}
```
Infinite Loops

```cpp
int x, y;
for(x = 0; x < 10; x++) {
    for(y = 0; y < 5; x++) {
        cout << "The value of x is: " << x << endl;
    }
}
```
Infinite Loops

```c++
int x, y;
for(x = 0; x < 10; x++) {
    for(y = 0; x < 5; y++) {
        cout << "The value of x is: " << x << endl;
    }
}
```
Infinite Loops

```cpp
int x;
for(x = 1; x <= 10; x++) {
    cout << "The value of x is: " << --x << endl;
}
```
The while loop

```cpp
for(x=1; x <= 100; x++)
    cout << "hello world\n";
```

VS.

```cpp
int x=1;
while(x<=100) {
    cout << "hello world\n";
    x++;
}
```
Common Mistakes

```c
int x=1;
while(x<=100) {
    cout << "hello world\n";
    x++; // What if we forget this?
}
```

What if we forget this?
The do/while loop

```cpp
int x=1;
do {
    cout << "hello world\n";
    x++;}
while(x<=100);
```

• Difference b/w while and do/while?
Infinite Loop Example...