CS 161
Intro to CS I

Repetition
New Operators for Expressions

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

• What if you have the statement `var = var - 5;` 
  OR `var = var + var;`

  `var -= 5;` 
  `var += var;`

• Pre vs. Post increment: `++var vs. var++`
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

```java
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

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

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

Test is False: Execution after loop
The while loop

```cpp
for(x=1; x <= 100; x++)
    cout << “hello world” << endl;
```

**VS.**

```cpp
int x=1;
while(x<=100) {
    cout << “hello world” << endl;
    x++;
}
```
Common Mistakes

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

What if we forget this?
The do/while loop

```c++
int x=1;
do {
    cout << "hello world" << endl;
    x++;
} while(x<=100);
```

• Difference b/w while and do/while?
Looping ...

• for loops
  – Repeat for specific number of times
  – Example?

• while loops
  – Repeat while a condition is being met
  – Example?

• do while loops
  – Always do once, and repeat while condition is met
  – Example?
The for Loop Examples

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

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

• 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

```c++
for(x = 0; x < 10; x++) {
    for(y = 0; y < 10; y++) {
        cout << "hello world" << endl;
    }
}
```

• 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

```cpp
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

```cpp
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;
}
```
Why is this good/bad?

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

- Where can we access x and y?
Demo...