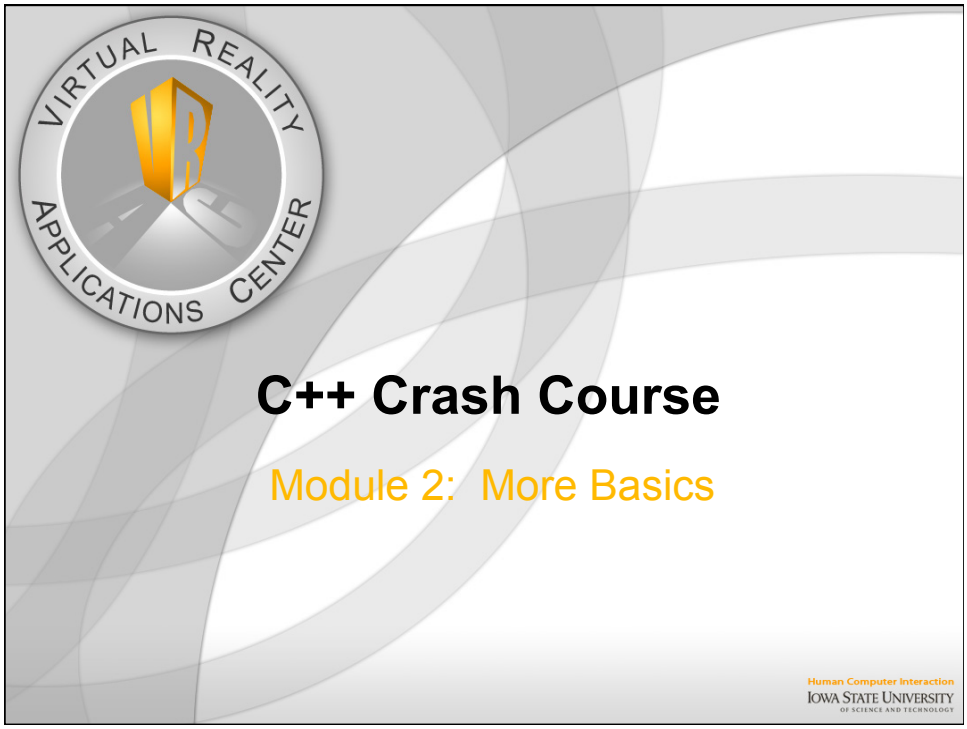
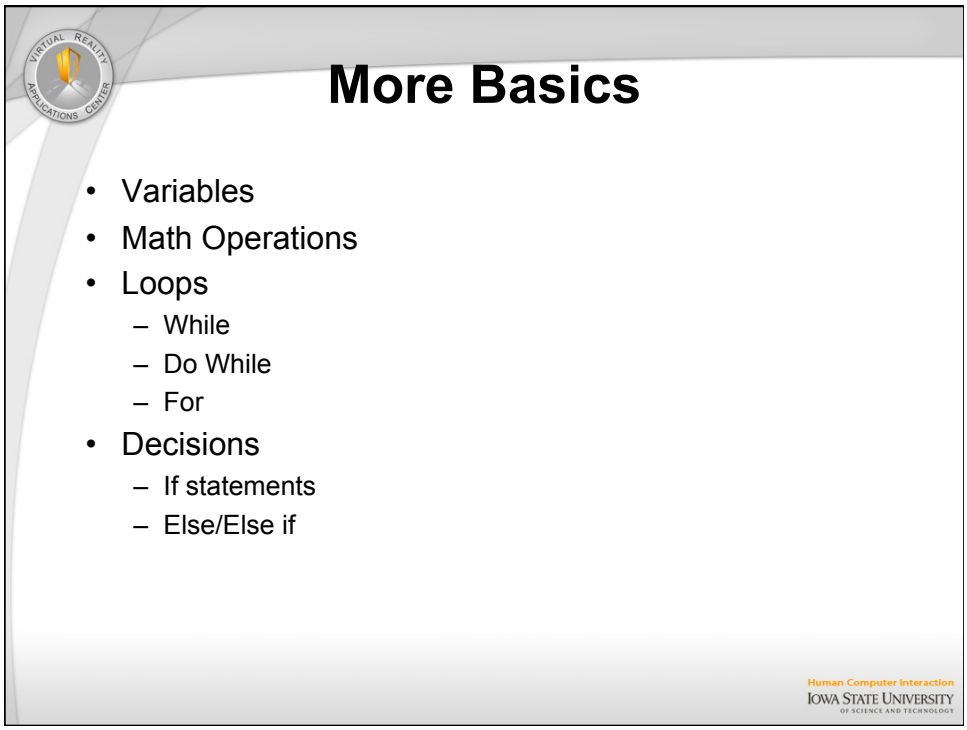
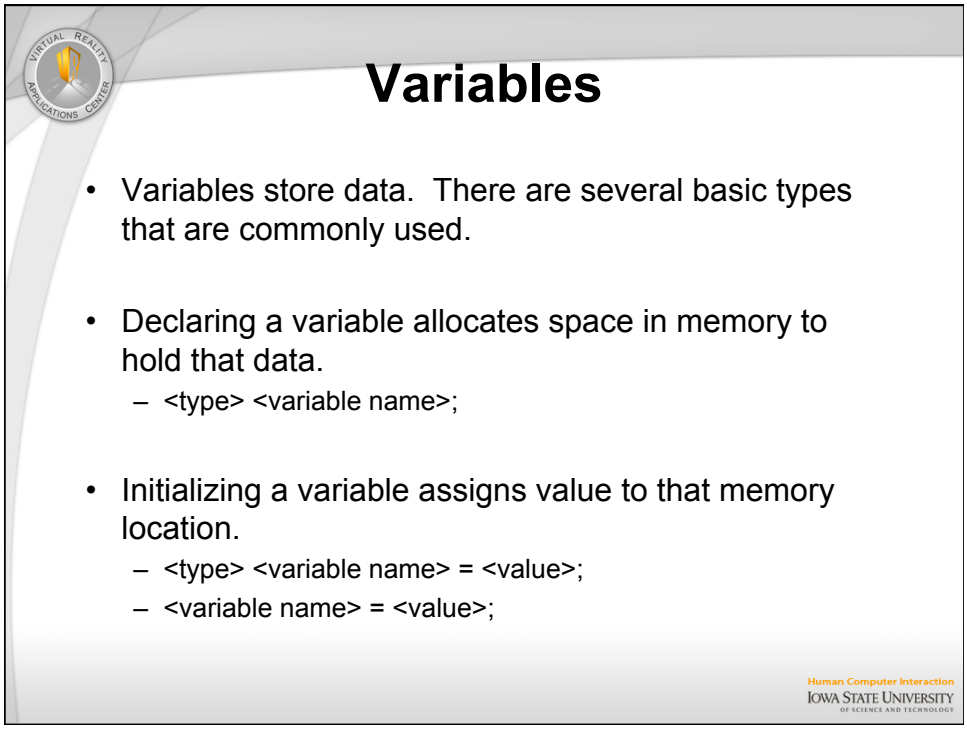
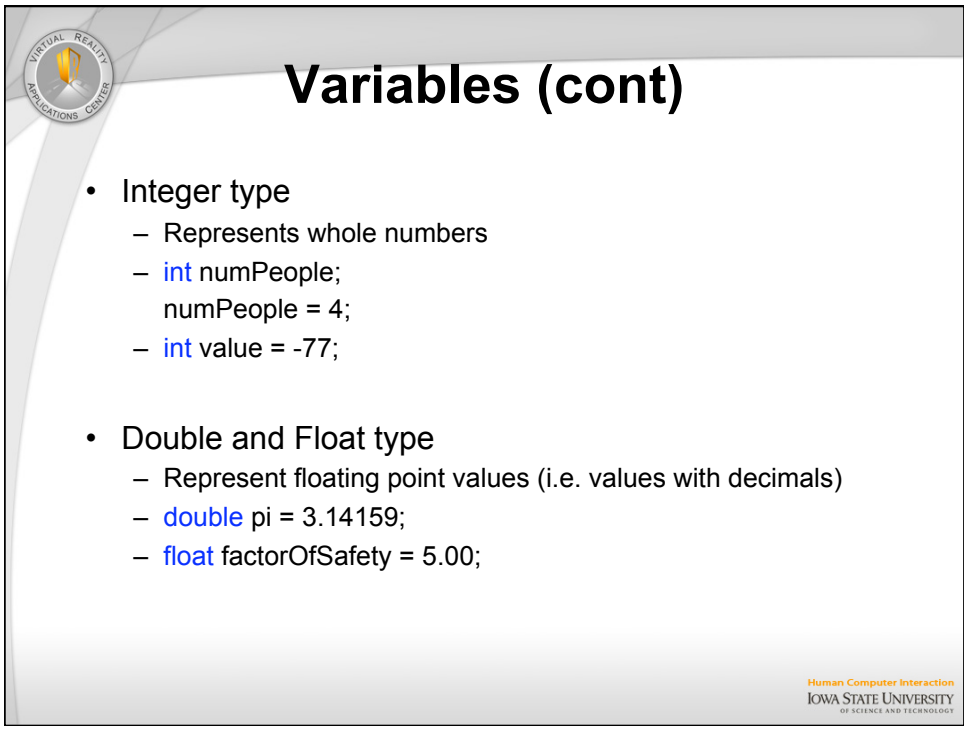
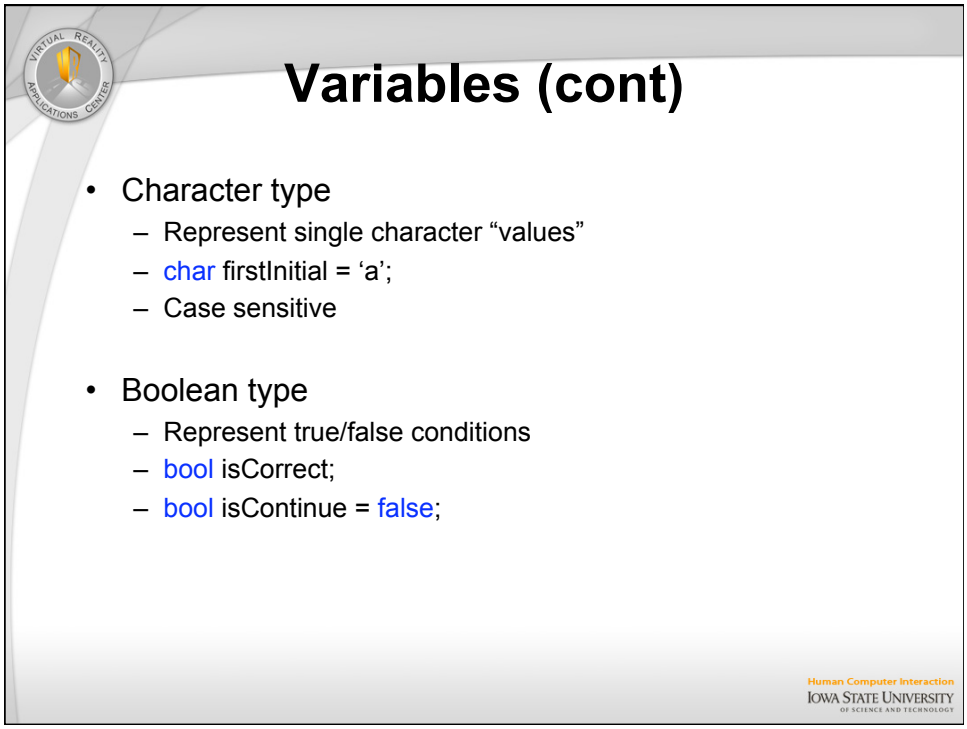
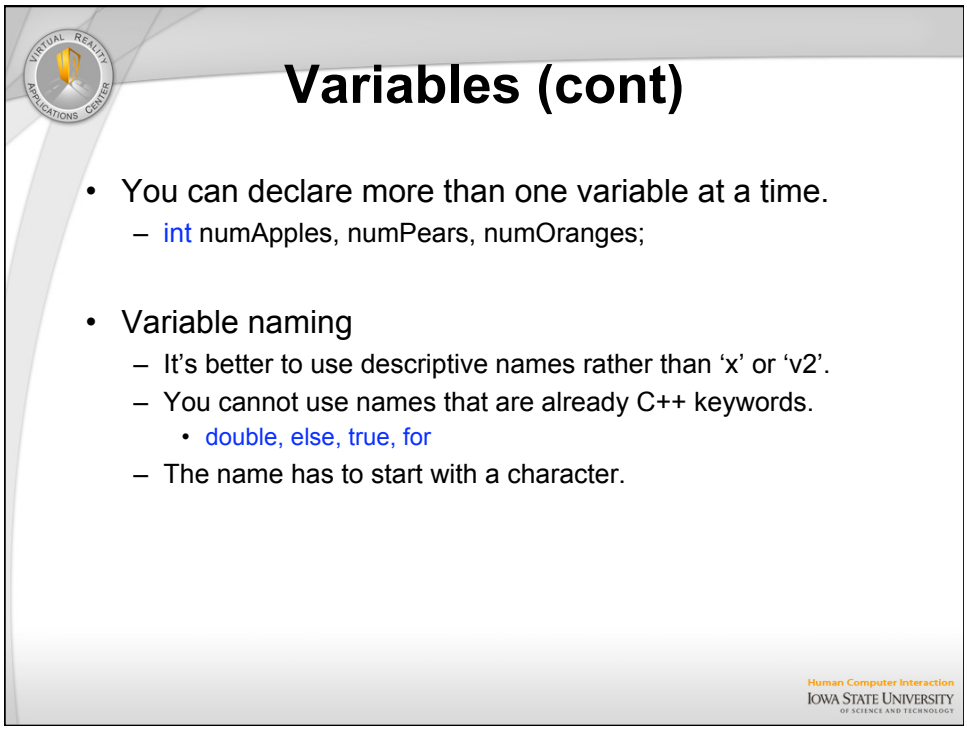

VIRTUAL REALITY
APPLICATIONS CENTER
C++ Crash Course
Module 2: More Basics
Human Computer Interaction
IOWA STATE UNIVERSITY
OF SCIENCE AND TECHNOLOGY


More Basics
• Variables
• Math Operations
• Loops
– While
– Do While
– For
• Decisions
– If statements
– Else/Else if
Human Computer Interaction
IOWA STATE UNIVERSITY
OF SCIENCE AND TECHNOLOGY


Variables
• Variables store data. There are several basic types that are commonly used.
• Declaring a variable allocates space in memory to hold that data.
– <type> <variable name>;
• Initializing a variable assigns value to that memory location.
– <type> <variable name> = <value>;
– <variable name> = <value>;
Human Computer Interaction
Iowa State University


Variables (cont)
• Integer type
– Represents whole numbers
– int numPeople;
numPeople = 4;
– int value = -77;
• Double and Float type
– Represent floating point values (i.e. values with decimals)
– double pi = 3.14159;
– float factorOfSafety = 5.00;
Human Computer Interaction
Iowa State University


Variables (cont)
Character type
Represent single character “values”
char firstInitial = ‘a’;
Case sensitive
Boolean type
Represent true/false conditions
bool isCorrect;
bool isContinue = false;
Human Computer Interaction
Iowa State University


Variables (cont)
You can declare more than one variable at a time.
int numApples, numPears, numOranges;
Variable naming
It’s better to use descriptive names rather than ‘x’ or ‘v2’.
You cannot use names that are already C++ keywords.
double, else, true, for
The name has to start with a character.
Human Computer Interaction
Iowa State University

Math Operations

- Addition: '+'
- Subtraction: '-'
- Multiplication: '*'
- Division: '/'
- Remainder: '%'
- Increment: '++'
- Decrement: '--'

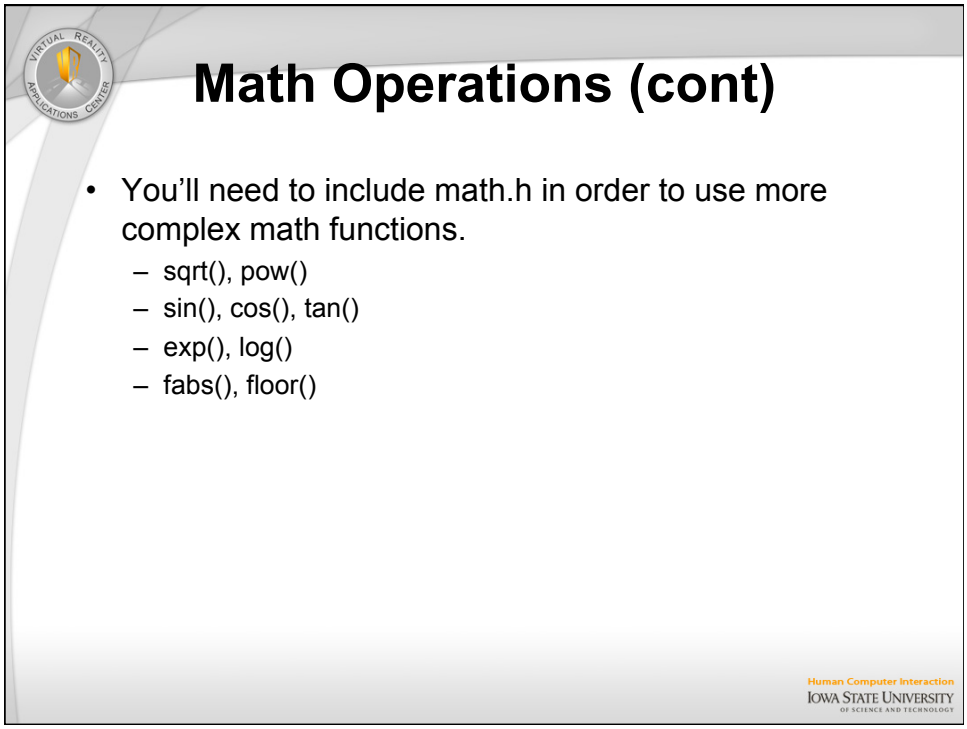
- Precedence
  - The order in which operations are carried out.

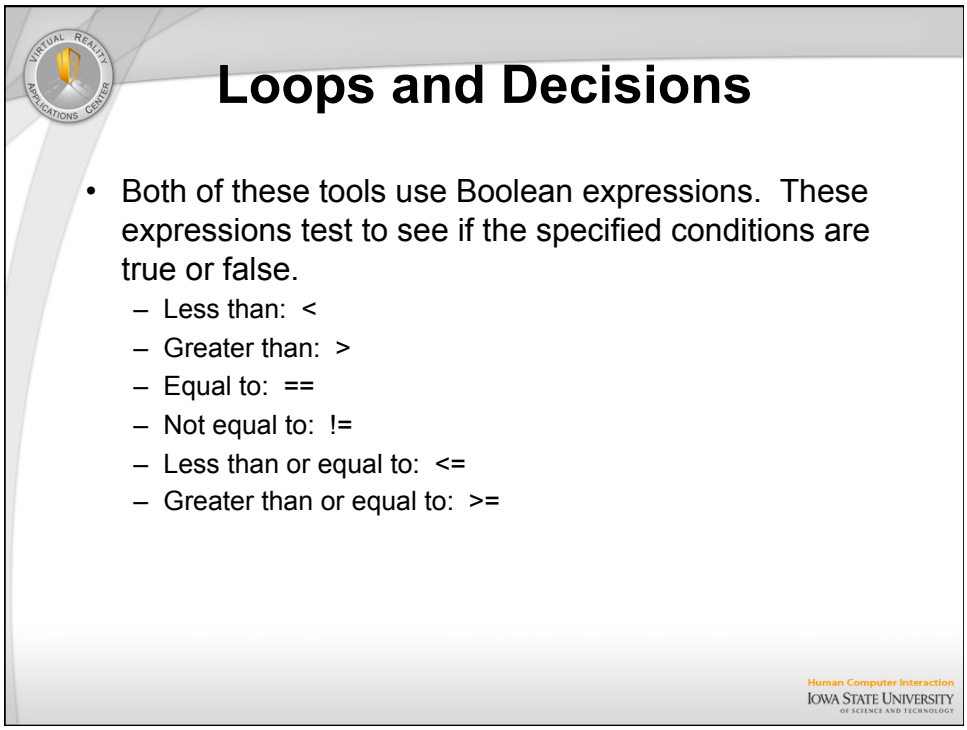
Parentheses	( )
Positive/Negative sign	+ -
Increment/Decrement	++ --
Operational Assignment	+= -= *= /= %=
Multiplicative	% * /
Addition/Subtraction	+ -
Assignment	=

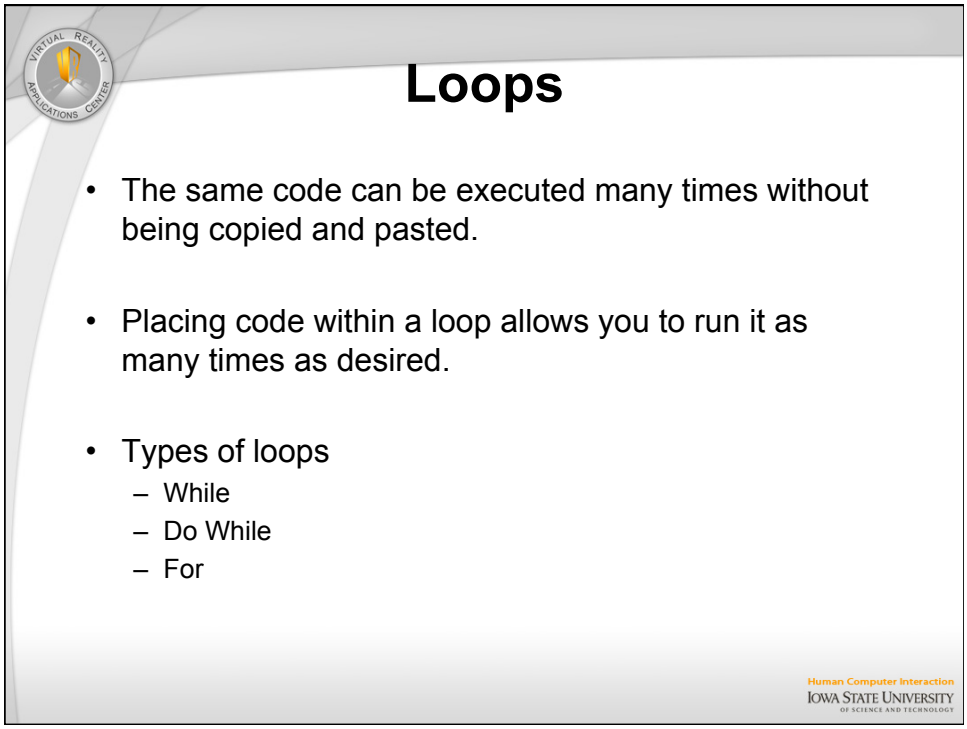
Math Operations (cont)

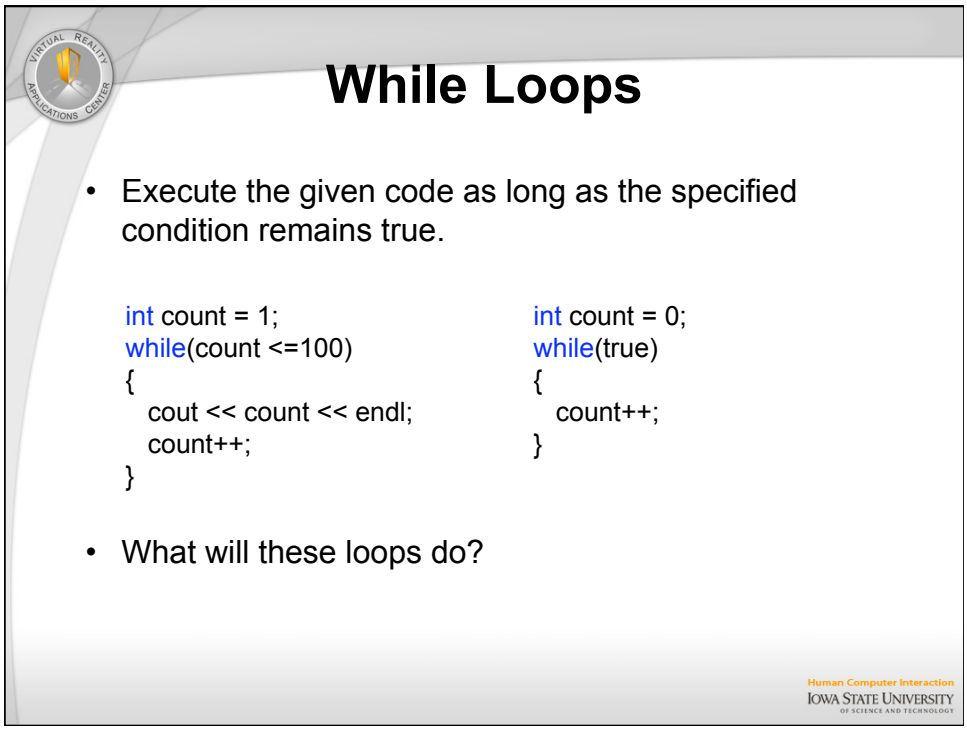
- Be careful when performing operations with multiple types.
  - int x = 5;
    double y;
    y = x/2;
  - Result? y = 2
  - ???

- Solutions
  - y = ((double) x)/2;
  - y = x/2.0;


Math Operations (cont)
• You'll need to include math.h in order to use more complex math functions.
– sqrt(), pow()
– sin(), cos(), tan()
– exp(), log()
– fabs(), floor()
Human Computer Interaction
Iowa State University
of Science and Technology


Loops and Decisions
• Both of these tools use Boolean expressions. These expressions test to see if the specified conditions are true or false.
– Less than: <
– Greater than: >
– Equal to: ==
– Not equal to: !=
– Less than or equal to: <=
– Greater than or equal to: >=
Human Computer Interaction
Iowa State University
of Science and Technology


Loops
• The same code can be executed many times without being copied and pasted.
• Placing code within a loop allows you to run it as many times as desired.
• Types of loops
– While
– Do While
– For
Human Computer Interaction
Iowa State University
of Science and Technology


While Loops
• Execute the given code as long as the specified condition remains true.
int count = 1;
while(count <=100)
{
cout << count << endl;
count++;
}
int count = 0;
while(true)
{
count++;
}
• What will these loops do?
Human Computer Interaction
Iowa State University
of Science and Technology

Do While Loops

- Very similar to a while loop. The check is just performed after each iteration.

```
int count = 1;
do
{
   cout << count << endl;
   count++;
} while(count <=100)
```

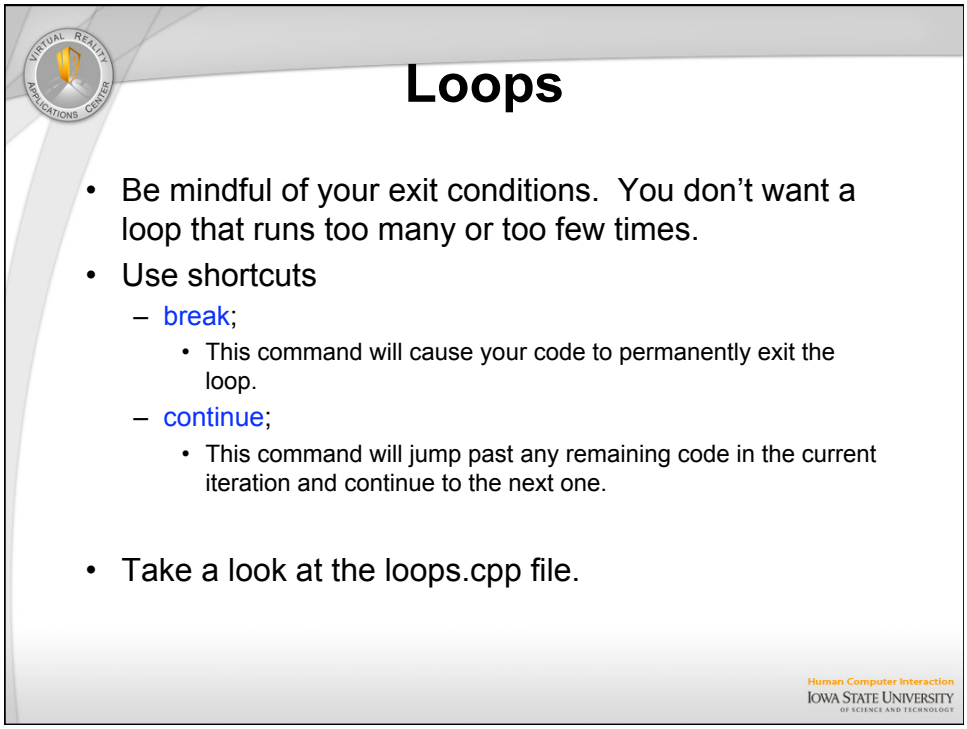
```
bool isContinue = false;
do
{
   count = 0;
}while(isContinue)
```

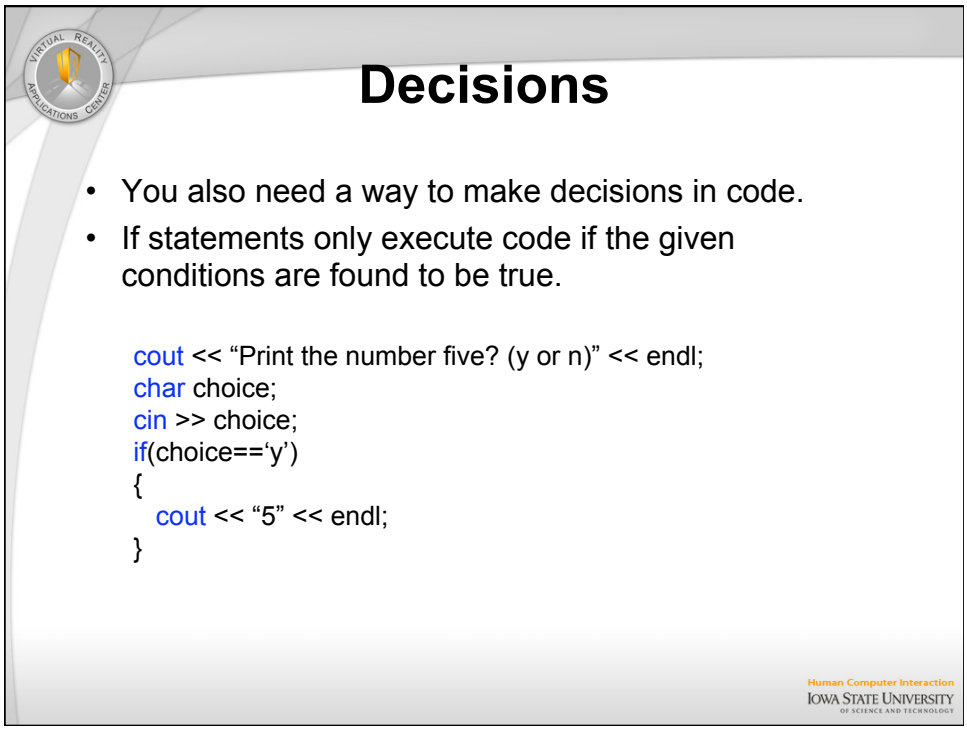

For Loops

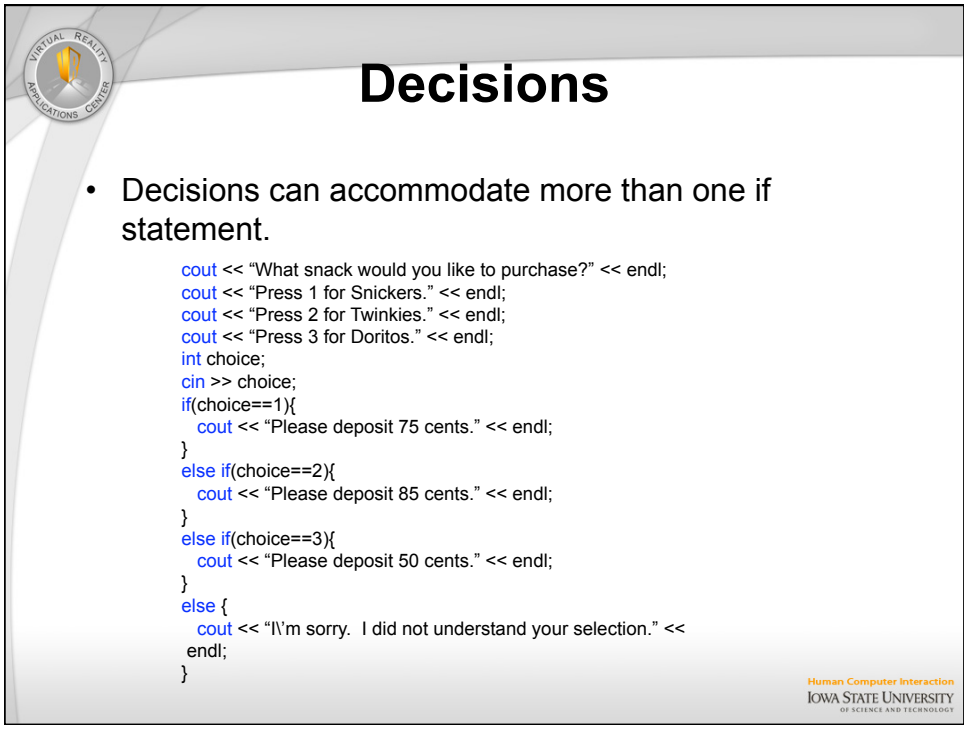
- Most often used when you know exactly how many iterations you want to run.
- Usage
  - for(start condition; end condition; increment)

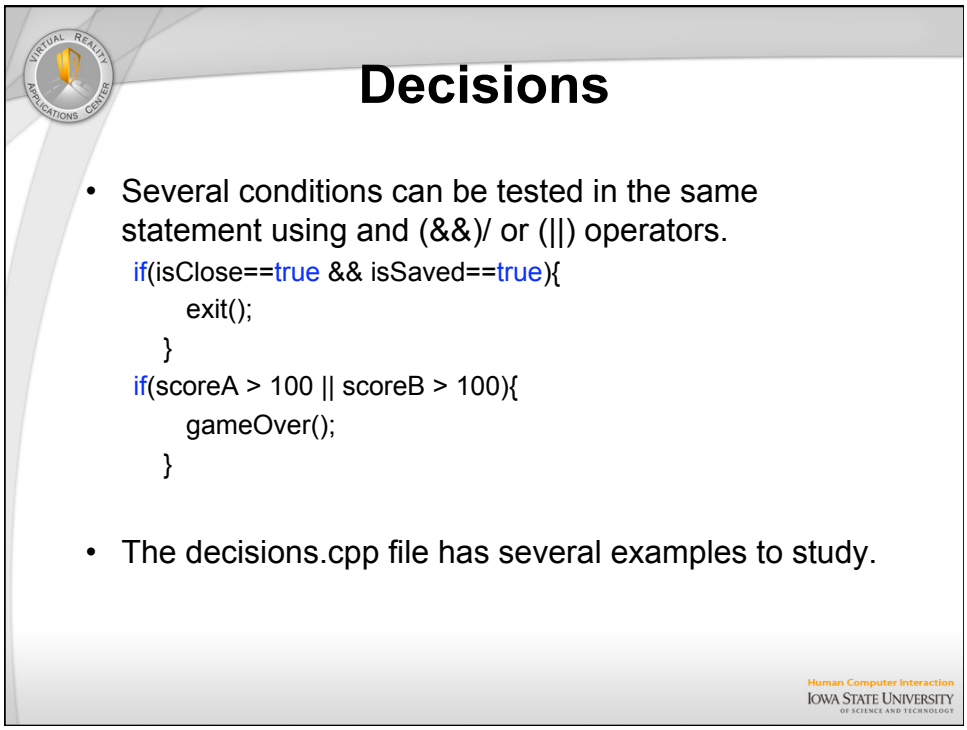
```
for(int i=0; i<10; i++)
{
  cout << i << endl;
}
```

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


Loops
• Be mindful of your exit conditions. You don't want a loop that runs too many or too few times.
• Use shortcuts
– break;
• This command will cause your code to permanently exit the loop.
– continue;
• This command will jump past any remaining code in the current iteration and continue to the next one.
• Take a look at the loops.cpp file.
Human Computer Interaction
Iowa State University


Decisions
• You also need a way to make decisions in code.
• If statements only execute code if the given conditions are found to be true.
cout << "Print the number five? (y or n)" << endl;
char choice;
cin >> choice;
if(choice=='y')
{
cout << "5" << endl;
}
Human Computer Interaction
Iowa State University


Decisions
• Decisions can accommodate more than one if statement.
cout << “What snack would you like to purchase?” << endl;
cout << “Press 1 for Snickers.” << endl;
cout << “Press 2 for Twinkies.” << endl;
cout << “Press 3 for Doritos.” << endl;
int choice;
cin >> choice;
if(choice==1){
cout << “Please deposit 75 cents.” << endl;
}
else if(choice==2){
cout << “Please deposit 85 cents.” << endl;
}
else if(choice==3){
cout << “Please deposit 50 cents.” << endl;
}
else {
cout << “I\’m sorry. I did not understand your selection.” <<
endl;
}
Human Computer Interaction
Iowa State University


Decisions
• Several conditions can be tested in the same statement using and (&&)/ or (||) operators.
if(isClose==true && isSaved==true){
exit();
}
if(scoreA > 100 || scoreB > 100){
gameOver();
}
• The decisions.cpp file has several examples to study.
Human Computer Interaction
Iowa State University