

Iteration Behavior

• Repeated execution

– Of the same set of statements



• C++ loop constructs

- while
- do/while
- for

Loop Construct: while

• Loop body

- Set of statements to execute
 - Zero or more times

• Loop condition

- Tested before each iteration
- Body executes as long as it's true

Loop Example: while

```

int x = 6;
const int firstInt = 1;
int upperLimit = x;
int oddSum = 0;
int nextOdd = firstInt;

while (nextOdd <= upperLimit)
{
  oddSum += nextOdd;
  nextOdd += 2;
}

```

Loop condition

Loop body

Loop Trace: while

```

int x = 6;
const int firstInt = 1;
int upperLimit = x;
int oddSum = 0;
int nextOdd = firstInt;
while (nextOdd <= upperLimit)
{
  oddSum += nextOdd;
  nextOdd += 2;
}
nextOdd: 1 3 5 7
oddSum: 0 1 4 9
upperLimit: 6 6 6 6

```

Loop Issues: while

- **Condition must sometime be false**
 - Or else we have an infinite loop
 - Loop never exits!
- **If condition false at beginning**
 - Loop body is never executed!

Loop in an Input Function

```

int InputNonNegInt ()
{
  int num;
  cout << "Enter an integer: ";
  cin >> num;
  while (num < 0)
  {
    cout << "Can't be negative: ";
    cin >> num;
  }
  return num;
}

```

Input value

Iterate until value is OK

⋮

Constructing Loops

- **Process a sequence**
 - Of values, data objects, etc.
- **Iterator object**
 - Knows the desired sequence
 - Startup - Set up to process first item
 - Next - Get next item in sequence
 - More - True if there are more items



Handwritten notes area with horizontal lines

⋮

Loop Structure: while

```

initialize
startup
while (more)
{
    process current item
    next
}
finalize

```



Handwritten notes area with horizontal lines

⋮

Loop Invariants 

- **Preconditions**
 - True immediately before loop
- **Loop invariants**
 - True at entry and after each iteration
- **Postconditions**
 - True after last execution

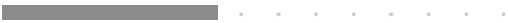


Handwritten notes area with horizontal lines

⋮

Loop Design Example

- **Find average of grades**
 - Positive integer values
 - Input by user
- **Unknown number of grades**
- **Sentinel value**
 - User enters negative number at end

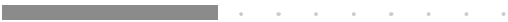


Handwriting lines for notes corresponding to the 'Loop Design Example' section.

⋮

Loop Design

- **Data objects**
 - sum - total of values
 - count - number of values
 - val - current input value
- **Loop condition**
 - val must be positive

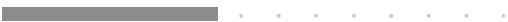


Handwriting lines for notes corresponding to the 'Loop Design' section.

⋮

Preconditions 

- **sum is zero**
- **count is zero**
- **val is current input value**



Handwriting lines for notes corresponding to the 'Preconditions' section.

Loop Invariants

- **sum is sum of numbers so far**
- **count is how many numbers so far**
- **val is current input value**

Consistent with preconditions

Postconditions

- **sum is sum of numbers so far**
- **count is how many numbers so far**
- **val is current input value**
- **val is negative (condition false)**

Consistent with loop invariants

Design Example Code

```

double avg = 0.0;
int sum = 0;
int count = 0;
int val = 0;
cin >> val;
while (val >= 0)
{
    sum += val;
    count++;
    cin >> val;
}
if (count > 0)
    avg = double(sum)/count;

```

Design Exercise



- **Accept a series of values**

- Positive integers
- Stop if negative number is entered

- **Print out divisors of each value**

- Starting with one
