# Basic **sed** by Examples

Ned Nedialkov

McMaster University
Canada

SE 3F03
January 2013

# Outline

# Examples

**sed** `-e 's/oldstring/newstring/g' infile > outfile`

- ▶ `-e` means expression
- ▶ `s` means substitute
- ▶ replaces `oldstring` by `newstring` in `infile`
- ▶ `g` means global: replace all occurrences
- ▶ `>` the output is redirected to `outfile`

- **sed** `-e '1,8d' infile`
  removes lines with numbers 1 to 8
- **sed** `-e '/include/d' infile`
  removes the lines that contain `include`
- **sed** `-e '/^FIRST/,/^LAST/d' infile`
  removes the lines between a line starting with `FIRST` and
  a line starting with `LAST`

# Processing cycle

**sed**
- ▶ reads a line from a file into a buffer
- ▶ executes the command (or commands) on the buffer
- ▶ outputs the buffer to the standard output

# Substitution

- **sed** -e 's/pattern1/pattern2/'
- **sed** -e 's/pattern1/pattern2/g'
- pattern1 text to be replaced; can be a regular expression
- Try
  sed -e 's/^\([0-9]\{3\}\)\-/(\1)-/' sedphones.txt
  on the file sedphones.txt containing

  ```
  905-123-3456
  905-124-3456
  905-125-3456
  905-126-3456
  905-127-3456
  ```