Max and min
Consider the quadratic function
\[ q(x) = x^2 + bx + c \]
on the interval \([L, R]\):
What is the minimum value of \(q(x)\) in \([L, R]\)?

Questions
- What are the critical points?
  - End points: \(x = L, x = R\)
  - \(\{x | q'(x) = 0\}\)
    \[ q(x) = x^2 + bx + c \]
    \[ q'(x) = 2x + b \]
    \[ q'(x_c) = 0 \Rightarrow x_c = -\frac{b}{2} \]
- What to do with the critical points?
Algorithm
- If \( x_c \in [L, R] \)
  - Min value is \( q(x_c) \)
- Otherwise
  - Min value is minimum of \( q(L) \), \( q(R) \)

Detailed Algorithm
- Calculate \( x_c \)
  - If \( L \leq x_c \leq R \)
    - Calculate \( q_{\text{Min}} \)
  - Otherwise
    - Calculate \( q_L = q(L) \), \( q_R = q(R) \)
      - If \( q_L < q_R \)
        - \( q_{\text{Min}} = q_L \)
      - Otherwise
        - \( q_{\text{Min}} = q_R \)

---

Conditional Statement

```java
if ( condition1 )
  statement1;
else
  statement2;
```

---

Conditional Statement

```java
if ( condition1 ) {
  statement1;
  statement2;
} else {
  statement3;
  statement4;
}
```

---

Conditional Statement

```java
if ( condition1 )
  statement1;
else if ( condition2 )
  statement2;
else
  statement3;
```

---

Conditional Statement

```java
if ( condition1 )
  statement1;
else
  if ( condition2 )
    statement2;
  else
    statement3;
```
Boolean Expressions
- A condition often uses relational and logical operators, which return boolean results:
  - `==` equal to
  - `!=` not equal to
  - `<` less than
  - `>` greater than
  - `<=` less than or equal to
  - `>=` greater than or equal to
  - `!` not
  - `&&` and
  - `||` or

User input
- Java5: standard class Scanner
- Java1.4.2: class Keyboard (by L&L)
- Examples:
  - `int i = Keyboard.readInt();`
  - `double d = Keyboard.readDouble();`
  - `char c = Keyboard.readChar();`
  - `boolean b = Keyboard.readBoolean();`

Example: k factorial
- Write a program fragment for calculating k!
- Use a while loop; assume k is given
- E.g., 4! = 1*2*3*4

Example: k factorial
- Write a program fragment for calculating k!
- Use a while loop; assume k is given
- `// k! where k>=0`
  - `int kfact= 1; // k!, initially 1!`
  - `int i= 2;`
  - `while ( i <= k ) {`
  - `  kfact= kfact*i;`
  - `  i= i + 1;`
  - `}`

The while loop
- Syntax:
  - `while ( condition )
    statement ;`
- Pattern for doing something n times
  - `int i= 1;`
  - `while ( i <= n ) {`
  - `  // do something`
  - `  // increment counter`
  - `  i= i + 1;<=`
  - `}

Shortcut expressions
- Increment operator
  - `i = i + 1; ⇒ i++;`
- Decrement operator
  - `i = i - 1; ⇒ i--;`

Use them on their own—not as part of another statement!
More shortcut expressions

- Assignment operators
  
  \[
  \begin{align*}
  s &= s + \text{val}; & \Rightarrow & \quad s &= += \text{val}; \\
  s &= s - \text{val}; & \Rightarrow & \quad s &= -= \text{val}; \\
  s &= s * \text{val}; & \Rightarrow & \quad s &= *= \text{val}; \\
  s &= s / \text{val}; & \Rightarrow & \quad s &= /= \text{val}; \\
  \end{align*}
  \]

Example: \text{Eeeeeeeeeee}

- Approximation is “better” with more terms
- Use \text{Math.exp()} for “true” value of \( e^x \)
- \text{Error: } |\text{true} - \text{approx}| 
- \text{Tolerance: } \text{tolerable amount of error}
- How many terms in the series do we need to have error \( < 0.0001 \) ?
- User inputs \( x \) value

\[
e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots + \frac{x^n}{n!} + \ldots
\]