
Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```


The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
$>$	Greater than
$<$	Less than
$>=$	Greater than or equal to
$<=$	Less than or equal to
$==$	Equal to
$!=$	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```


Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```


Relational Operators

Operator	Meaning
$>$	Greater than
$<$	Less than
$>=$	Greater than or equal to
$<=$	Less than or equal to
$==$	Equal to
$!=$	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```


Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
$>$	Greater than
$<$	Less than
$>=$	Greater than or equal to
$<=$	Less than or equal to
$==$	Equal to
$!=$	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```


The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```


Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```


Relational Operators

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
$>$	Greater than
$<$	Less than
$>=$	Greater than or equal to
$<=$	Less than or equal to
$==$	Equal to
$!=$	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
$>$	Greater than
$<$	Less than
$>=$	Greater than or equal to
$<=$	Less than or equal to
$==$	Equal to
$!=$	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```


Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```


The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
$>$	Greater than
$<$	Less than
$>=$	Greater than or equal to
$<=$	Less than or equal to
$==$	Equal to
$!=$	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
$>$	Greater than
$<$	Less than
$>=$	Greater than or equal to
$<=$	Less than or equal to
$==$	Equal to
$!=$	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```


Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```


Relational Operators

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
$>$	Greater than
$<$	Less than
$>=$	Greater than or equal to
$<=$	Less than or equal to
$==$	Equal to
$!=$	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```


Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?

Relational Operators and the If Statement

Conditionals

- So far, we can Input, Output and Calculate
- How can we explore relationships between data?
- How can our program only do things *sometimes*?

Decisions!

- Relational Expressions allow our program to *make a decision*
 - based on the **data** in the program
- What are some decisions we might want our program to make?

Relational Expression

- Relational expression: an expression that uses a **Relational Operator**
 - its value is a **Boolean value** (True or False)

```
int height = 32;
```

```
int minHeight = 34;
```

```
bool bVal;
```

```
bVal = height > minHeight;
```

```
bVal = height == minHeight;
```

```
bVal = height >= (minHeight - 2);
```

Relational Operators

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

- All are binary operators
- Left to right associativity

Precedence

Precedence Operators (Highest to Lowest)

(unary negation) -

* / %

Arithmetic Operators

+ -

> >= < <=

Relational Operators

== !=

= += -= *= /= %=

Assignment Operators

const Declarations

- Constant declaration

```
const double PI = 3.14;
```

```
const double RADIUS = 5.4;
```

- Constant declarations are fixed and cannot be changed
- By convention, constants are always UPPERCASE

Practice

- What is the value of the following Relational Expressions?

Relational Operators work on Integers, Floating point numbers, and Characters.

```
int width = 99;
```

```
const int HEIGHT = 42;
```

```
bool bVal;
```

```
bVal = width > HEIGHT;
```

```
bVal = -width <= HEIGHT;
```

```
bVal = width != HEIGHT;
```

```
bVal = width == (width + 1);
```

```
bVal = width == width + 1;
```

The `if` Statement

- We execute each statement in our program in order.
- What `if` we only want to execute a statement sometimes?
- The `if` Statement!

```
if ( condition )  
{  
    //statements  
}
```

Practice: What is the output?

```
int age;
const int VOTING_AGE = 18;

cin >> age;

if ( age >= VOTING_AGE )
{
    cout << age << " > "
        << VOTING_AGE;
    cout << " You can vote!"
        << endl;
}
```

Practice

- For the problem below:
 - what data will you need?
 - what will you need to do conditionally?
 - what data will you use in your decision?
- Calculate the average grade for all three exams in a course. Print a message stating if the student passed the course.

“You passed!”

“You failed!”

Example

- Your local bookstore has asked you to write a program to help them determine the cost of shipping of customer orders. If the order is \$30 or less then shipping will cost \$5, if the order is over \$30 then shipping will be \$3.
- Write the program to solve this problem

Problem

- The bookstore has now changed its shipping policy so that
 - If the order is \$30 or less, shipping is \$5
 - If the order is over \$30 but less than \$50, shipping is \$3
 - If the order is \$50 or more then shipping is \$2
- What would we need to change in the program?