

# Mixing Types

## Lecture 6

### Sections 3.2, 3.3

Robb T. Koether

Hampden-Sydney College

Fri, Sep 7, 2018

- 1 Mixing Data Types
  - Promotion
  - Conversion
- 2 Assignment Statements
- 3 Mixed Expressions
- 4 Assignment

# Outline

## 1 Mixing Data Types

- Promotion
- Conversion

## 2 Assignment Statements

## 3 Mixed Expressions

## 4 Assignment

# Outline

- 1 Mixing Data Types
  - Promotion
  - Conversion
- 2 Assignment Statements
- 3 Mixed Expressions
- 4 Assignment

# Mixing Types

- We can write numerical expressions involving objects of different types, but the processor can perform the calculation only if the objects are of *exactly* the same type.
- What if they are not?
- Then one of them must be converted to the type of the other.

# Promotion of Types

- **Promotion** is used in expressions that contain objects of different *types*, but in the same *family* (integer family, floating-point family).
- Promotion never alters the value of the object; it affects only the amount of memory occupied by the object.
- The “smaller” type is converted to the “larger” type with no change in value.

# Promotion of Types

- Integer types are promoted as

**char** → **short** → **int** → **long** → **long long**

- Floating-point types are promoted as

**float** → **double** → **long double**

- The result is of the type of the object that uses more bytes of memory.

# Promotion of Types

## Promotion of Types

```
short a = 123;  
int b = 456789;  
cout << a + b << endl;  
  
float r = 1.2f;  
double pi = 3.14159265358979;  
cout << pi*r*r << endl;
```

- The **short** 123 is promoted to **int**.
- The **float** 1.2f is promoted to **double**.



# Outline

- 1 Mixing Data Types
  - Promotion
  - Conversion
- 2 Assignment Statements
- 3 Mixed Expressions
- 4 Assignment

# Conversion of Types

- A **type conversion** occurs when objects from two *different* families of types are used in the same statement.
- This typically involves a mixture of an integer type and a floating-point type.
  - The integer type is converted to a floating-point type.
- The two important cases of this are
  - Mixed assignments.
  - Mixed expressions.

# Outline

- 1 Mixing Data Types
  - Promotion
  - Conversion
- 2 Assignment Statements**
- 3 Mixed Expressions
- 4 Assignment

# Assignment Statements

## Assignment Statements

```
int a = 10;      // Initialization
int b = 20;      // Initialization
    ⋮
b = 3*a + 1;     // Assignment
```

- An **assignment statement** is a statement of the form  
$$variable = expression;$$
- The value of the expression is calculated and then assigned to the variable.
- That becomes the new value of the variable; the old value is gone.

# Mixed Statements

## Mixed Assignments

```
int a = 10;  
float x = 2*a + 20;  
double y = x + 5.0f;
```

- A **mixed assignment** is an assignment statement in which the type of the object on the left is different from the type of the expression on the right.
- The type of the value on the right must be promoted or converted to the type of the object on the left.

# Mixed Statements

- Conversion may result in a loss of precision or a significant change in value.
- Conversion may not be possible, resulting in an error message.

# Outline

- 1 Mixing Data Types
  - Promotion
  - Conversion
- 2 Assignment Statements
- 3 Mixed Expressions**
- 4 Assignment

# Mixed Expressions

## Mixed Expressions

```
int a = 10/3.0f;
```

- A **mixed expression** is an expression that includes objects from two different families of types.
- This typically involves a mixture of integer types and floating-point types.
- Both objects are converted to **double**.
- The operation is performed on **doubles**.
- The result is a **double**.



# Type Casting

## Type Casting

```
int sum = 0;      // The sum of the integers
int count = 0;    // The number of integers
:
float avg = (float) sum/count;
```

## Type Casting

```
float feet;      // The length in feet
:
int yards = (int) (feet/3.0); // Whole number of yards
```

- A conversion of an expression type may be forced by using **type-casting**.
- To type-cast, write the new type in parentheses in front of the expression to be converted.

# Examples of Type Conversions

- Examples

- `Promotion.cpp`
- `MixedAssignments.cpp`
- `MixedExpressions.cpp`
- `BattingAverage.cpp`

# Outline

- 1 Mixing Data Types
  - Promotion
  - Conversion
- 2 Assignment Statements
- 3 Mixed Expressions
- 4 Assignment**

# Assignment

## Assignment

- Read Sections 3.2, 3.3