Type Casting

Section 3.3-3.5

Implicit Type Conversion (3.3)

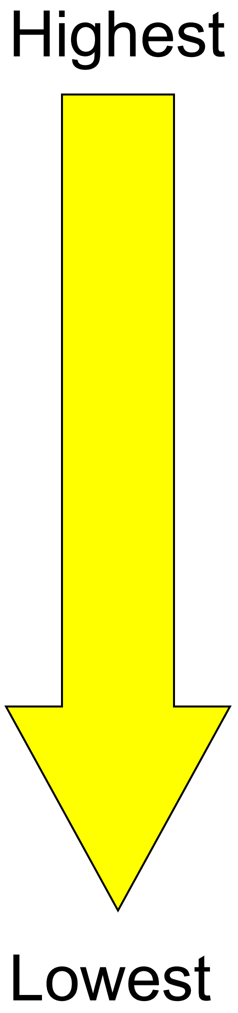
- Mixing the data types of operands during mathematical operations

```
const double PI = 3.14159265;
int intArea = PI * 4 * 4;
double doubleArea = PI * 4 * 4;

cout << intArea << " " << doubleArea;
```

Data Type Ranking

`long double`
`double`
`float`
`unsigned long`
`long`
`unsigned int`
`int (char, short, unsigned short)`



Rules for Implicit Type Conversion

- When a value is converted to a higher data type, it is being *promoted*
- When a value is converted to a lower data type, it is being *demoted*
  - Rule 1: **`char`**, **`short`**, and **`unsigned short`** are automatically promoted to **`int`** when used in a mathematical expression
  - Rule 2: When an operator works with values of different types, the lower ranking value is promoted to the higher ranking
  - Rule 3: When the value of an expression is assigned to a variable, it is converted to the data type of that variable

Q.2 Practice

- Assume the following variable definitions

```
int a = 5, b = 12;

double x = 3.4;
```

- What are the datatypes of the following expressions:

```
b / x

x * a

a % b * x
```

Explicit Type Conversion (3.4)

- A type cast expression let’s you manually change the data type of a value
- The syntax for type casting is
  - `static_cast<DataType>(Value)`
  - Value is a variable or literal value
  - DataType is the data type that you want to produce
  - Value is not altered

7.3 Example of Type Casting

```
double number = 3.7;
int val;
val = static_cast<int>(number);
```

- What is saved into val?

Uses of Type Casting

- Preventing integer division

```
int books = 30, months = 7;
double booksPerMonth;

booksPerMonth = static_cast<double>(books) / months;
```

- ASCII conversions

```
int number = 65;
cout << static_cast<char>(number);

cout << static_cast<int>('B');
```

Q.4 Practice

- What is the value of each of the variables while this expression is being executed?

```
int sum;
double gradeCounter, average;
sum = 324;
gradeCounter = 4;
average = static_cast<double>(sum) / gradeCounter;
```

Overflow and Underflow (3.5)

- What happens when a variable is assigned a value that is outside the datatype's range?

```
short testVar = 32767; // range for short?
cout << testVar << endl;
testVar = testVar + 1;
cout << testVar << endl;
testVar = testVar - 1;
cout << testVar << endl;
```

Combined Assignments

- The same variable can be used on the left hand side of the assignment and on the right hand side

```
notes = notes + 20;

notes = notes % 20;
```

- These are common in programming, so the two operators can be combined as follows:

```
notes += 20;

notes %= 20;
```

Examples of Combined Assignments

Operator	Example Usage	Equivalent To
`+=`	`x += 5;`	`x = x + 5;`
`-=`	`y -= 2;`	`y = y - 2;`
`*=`	`z *= 10;`	`z = z * 10;`
`/=`	`a /= b;`	`a = a / b;`
`%=`	`c %= 3;`	`c = c % 3;`

Q.5 Combined Assignments

- Combined assignments can be combined with arithmetic operators

```
y -= a * 2;

a /= b + c;

c %= d - 3;
```

- What is the long form of these statements?

Q.6 What is the Output?

```
int valOne=1, valTwo=2, valThree=3;


valOne += 4;
valTwo *= 2;
valThree -= 4;
valOne /= 3;
valTwo += valThree;
cout << valOne << endl;
cout << valTwo << endl;
cout << valThree << endl;
```

Multiple Assignments (3.7)

- C++ allows statements such as:

```
int a=0, b=1, c=2, d=3;
 a = b = c = d = 45;
```

- What value ends up in a?
  - Which assignment operator is performed first?
  - Why?

Q.1 Practice

- Write a C++ program that allows the user the ability to enter the number of nickels and pennies they have. The program should then display the corresponding number of whole dollars and left over change. The change should be in the form of the number of nickels and pennies.
  - data?
  - calculations?