Rawlston Morton

September 30, 2020

C# Ch 22

1. \*\*Explain the diﬀerence between the concepts of associativity and precedence.\*\*

Precedence is the order in which you read operators while the associativity is the direction (left => right or right => left) operator evaluates from.

2. \*\*Explain the diﬀerence between the concepts of binary and unary operators.\*\*

A unary operator is an operator that has just one operand (increment, ++) while a binary operator is an operator that has two operands (multiplication, \* ).

3. \*\*List four constraints imposed by C# with respect to operator overloading.\*\*

- You cannot change the precedence and associativity of an operator.

- You cannot change the multiplicity (the number of operands) of an operator.

- You cannot invent new operator symbols.

- You can’t change the meaning of operators when they are applied to built-in types.

4. \*\*What is the syntax for overloading operators? Discuss access, scope, return value types, and parameter types and multiplicity.\*\*

like a method but with the operator keyword

public static Hour operator +(Hour lhs, Hour rhs) => new Hour(lhs.value + rhs.value);

All operators must be public.

All operators must be static. The scope is limited to the objects of the class you defined it in.

Operators are never polymorphic and cannot use the virtual, abstract, override, or sealed modiﬁers.

5. \*\*What are symmetric overloaded binary operators and how do they diﬀer from non-symmetric overloaded binary operators?\*\*

Symmetric operators come in pairs (< > == !=)

6. \*\*Can you overload compound assignment operators? If so, please state how you do so. If not, explain why not.\*\*

Yes (implicitly), because the compound assignment operators while written like a @= b (where @ represents any valid operator) is actually read as a = a @ b.

7. \*\*What is the diﬀerence between overloading increment and decrement operators for value types and reference types?\*\*

Increment and decrement on reference types create new objects based on the data in the original. the data in the new object is incremented, but the data in the original is left unchanged

8. \*\*Why do we overload some operators in pairs?\*\*

The neither-or-both rule

9. \*\*What is the diﬀerence between widening conversion and narrowing conversion?\*\*

An implicit conversion is sometimes called a widening conversion because the result is wider than the original value—it contains at least as much information as the original value, and nothing is lost. An explicit conversion is sometimes called a narrowing conversion because the result is narrower than the original value (that is, it can contain less information) and can throw an OverﬂowException exception if the resulting value is out of the range of the target type.

10. \*\*What is the diﬀerence between explicit conversion and implicit conversion?\*\*

If a conversion is always safe, does not run the risk of losing information, and cannot throw an exception, it can be deﬁned as an implicit conversion. Otherwise, it should be declared as an explicit conversion.