Fold Expressions Solutions

Fold Expressions

- Explain briefly what is meant by a fold expression
  - A fold expression is used with the arguments to a variadic template
  - It tells the compiler to apply an operator to all the elements in the function parameter pack
- What are "left fold" and "right fold" expressions?
  - A left fold expression is evaluated from left to right
  - A right fold expression is evaluated from right to left

Fold Expression

- Complete the following variadic function template by adding a left fold expression that adds all arguments and returns the result

```
template <typename... Args>
auto add(Args... args) {
   return (... + args);
}
```

Fold Expression

- Before C++17, how were variadic template arguments processed?
  - The arguments were processed by recursive calls which remove an argument each time
  - A template specialization handles the case with one argument
- What are the advantages of using fold expressions?
  - Only one function required
  - The code more clearly expresses the programmer's intention
  - No recursion - the code compiles and executes faster

Fold Expression with Initial Value

- Alter your previous solution so that the initial value of the sum is 100
- Why was it not necessary to provide an initial value before?
  - For many operators, a default value is used
  - For addition and subtraction, this is 0

Printing Arguments

- Write a variadic template function that uses a fold expression to print out all its arguments
- Write a program to test your function

"Short-circuit" evaluation

- Explain what is meant by "short-circuit" evaluation
  - A logical operation will complete as soon as the result is known
  - e.g. if an argument to an AND operation evaluates as false, no more arguments will be processed
- Write a variadic template function with a fold expression that uses short-circuit evaluation
- Write a program to test your function