Lambda Expressions Introduction Solutions

Function call operator example

- Write a class with a function call operator
  - The function call operator takes one argument, of type int
  - The function call operator returns bool
  - The return value is true if the argument is an exact multiple of 2, otherwise false
- Write a program to test your class

The need for lambdas

- Describe some of the disadvantages of using functors to provide predicate functions for algorithms
  - Boilerplate code (several lines to implement a one-line function)
  - Needs a name which does not conflict with anything else in scope ("namespace pollution")
  - Interrupts writing and reading the code (productivity)

Lambda Expressions

- Explain briefly what is meant by a lambda expression
  - A lambda expression is an executable block of code that is written as a "local function", usually within another executable block of code
- Explain briefly how a lambda expression is implemented by the compiler
  - The compiler will generate a functor whose () operator contains the code from the body of the lambda expression
  - The compiler will create an instance of this functor and call it when required

Lambda Function Syntax

- When creating a lambda expression, how do we enter:
- Its name?
  - Lambda expressions are anonymous. We put [] where the name of the function would go
- Its arguments?
  - Inside (), just like a normal function
- Its function body?
  - We write the function body inside {}, just like we would for an inline function

Lambda Expression Return Type

- When creating a lambda expression, how do we enter its return type?
  - The return type will be automatically deduced, provided the lambda function body is a single return statement (C++11). Otherwise, it will be assumed to be void
  - This will be automatically deduced, provided the return type is the same in all paths through the function body (C++14)
  - This will be automatically deduced in all cases (C++17)
  - If we need to supply a return type, we add a trailing return type

```
[](int n) -> bool {
```

Example of lambda expression

- Rewrite your program from the first exercise to use a lambda expression instead of a functor