

Function Call Operator Solutions

Function call operator

- Briefly describe the function call operator
 - The function call operator is a member function called operator ()
 - The function call operator can take any number and type of arguments and return any type, as we wish
 - It allows us to create objects which can be called like functions
 - A class with a function call operator is called a functor
 - Functor objects can be passed as arguments to other functions and returned from function calls, like any other objects

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

Functors

- Explain what a functor is
 - In C++, a functor is a class with an overloaded function call operator
- What is a functor used for?
 - It is a class that behaves like a callable function
 - A functor makes a function into a "first class object"
 - It can be passed to function calls and returned from function calls, just like any other object

Function call operator example

- Explain what the class below does
 - It has a function call operator which returns true if its argument is even, otherwise false
- Implement this class and write a program to exercise it

```
class evenp {  
    public:  
        bool operator() (int n) {  
            return (n % 2 == 0);  
        }  
};
```

Function Call Operator Example 2

- Write a function which takes a vector and an object of type `evenp` as argument
- The function passes each element of the vector as an argument to the object's function call operator
- It prints out the result of the call
- Write a program to test your function

Functors with state

- What does it mean for a functor to have state?
 - A functor with state has a data member (or members) which stores data between calls to the function call operator
- Repeat the previous exercises, but this time with a more general version of the functor
 - Instead of checking for multiples of 2, it checks for multiples of a number which is stored in a data member
 - The data member is initialized in the constructor
 - For example, we could create an object with this value as 3
 - Every time the function call operator is invoked, it checks whether its argument is divisible by 3