

Lambda Functions Part Four

Exercises

- What is meant by variable capture in a lambda expression?
- What options are available to a programmer who wants to be able to alter a captured variable?

- Write a simple program which shows the difference between default capture, mutable capture and capture by reference. Explain the results
- What considerations should a programmer bear in mind when capturing a variable by reference?

- What is meant by implicit capture?
- What is the syntax for performing an implicit capture?
- If we are performing an implicit capture by reference, how can we prevent the lambda function from modifying an important variable?

- Create a class which defines a lambda function in one of its member functions. The lambda function should modify a data member of the class
- Write a simple program to test your code

- Consider the code on the next slide
- After executing this code, which of the following gives the resulting values of x, y and z be?

x = 42, y = 99, z = 0

x = 42, y = 99, z = 141

x = 42, y = 99, z = 145

x = 43, y = 100, z = 143

x = 44, y = 101, z = 145

x = 44, y = 101, z = 141

- Write a program to check your answer. Explain the results

- This code creates a lambda expression, stores it in the variable `l` and evaluates it twice

```
int x{42}, y{99}, int z{0};
```

```
auto l = [=,&z]() mutable { ++x; ++y; z = x + y; };
```

```
l();
```

```
l();
```