Philip Pesic

Week 15

November 27 2022

Week 15 Lecture 15 Notes

Functions

Functions are a form of non-contiguous code reuse. They are declared in classes or outside of the main function, and they can have a number of return types and parameters.

Ex:

```
void function() {cout << “I’m a function!”;}
```

Function overloading

Function overloading is the use of multiple functions with the same name. These functions are differentiated from each other by their parameters. The function that is called will depend on the arguments that the call passes.

Ex:

```
Void func(int x) {}
Void func(float x) {}
Int main() {
func(3.14) //Calls second function, since 3.14 is a float
func(1) //Calls first function since 1 is an int
};
```

Operator overloading

Philip Pesic

Week 15

November 27 2022

Week 15 Lecture 15 Notes

Similar to function overloading, you can use the **operator** keyword to define what an operator will do for a specific function. The only operators that cannot be overloaded are: ::, .*, ., ?:

Ex:

```
ClassName operator opSymbol(args) {body}
```

```
MyClass operator - (MyClass m1, int x) {x = m1 - 5;}
```