Philip Pesic

Week 14

November 20 2022

Week 14 Lecture 14 Notes

Polymorphism

Polymorphism is the use of one piece of code to run several functions. To create a polymorphic function, first create a base class with a virtual function. Then, create a derived class, and create a function with the same name. This will inherit the properties of the base function, but it morphs the derived function into a new, unique function.

Ex:

```
Class Base {
Virtual void func() {cout << “Base”;}
};
Class Derived: public Base {
Void func() {cout << “Derived”;}
};
```

Philip Pesic

Week 14

November 20 2022

Week 14 Lecture 14 Notes

Virtual Functions

Virtual functions are functions which can be created in a base class that does not inherit from another class. They allow the function's behavior to be overwritten if a derived class declares a function with the same name. Virtual functions are an essential part of polymorphism, and OOP as a whole.

Ex:

```
Class Base {
Virtual void func() {cout << “Base”;} //Base Virtual function
};
Class Derived: public Base {
Void func() {cout << “Derived”;} //Same name - overrides base function
};
```

Philip Pesic

Week 14

November 20 2022

Week 14 Lecture 14 Notes

Abstract Base Class

An abstract base class is a base class whose virtual function has no body. This kind of function is also called a pure virtual function, and it serves no purpose in polymorphism.

Ex:

```
Class Abstract {
Virtual void func() {}
};
Class NonAbstract {
Virtual int func(int x) {return x;}
};
```