

# Virtual Functions Solutions

# Member Function Call

- Explain what is meant by static and dynamic binding when calling a member function on an object of a class
  - Static and dynamic binding refer to the two different ways in which a member function can be called on an instance
  - In static binding (the default), the static type of the object is used
  - In dynamic binding, the dynamic type is used

# Dynamic Binding Requirements

- Which two conditions must apply for dynamic binding to occur?
  - A member function is called on a pointer or reference to a base class
  - The member function was declared virtual in the base class

# Virtual Member Function Call

- What is a virtual member function?
  - A virtual member function has the virtual keyword before it
- What happens when a virtual function is called through a pointer or reference to the base class?
  - The compiler generates some code which the program uses at runtime to decide which function to call
  - The decision is based on the dynamic type of the object

# Function Call Argument

- Explain how virtual functions allow us to pass objects from a class hierarchy to a function which calls a member function
  - We declare the member function virtual in the base class
  - We pass the argument as a pointer or reference to the base class
- Write a program which calls such a function

# Extending the Hierarchy

- Extend the hierarchy from the previous exercise by adding a new derived class
- Call the function with an object of this class
- Explain your results
  - Dynamic binding means that the derived class's member function is called without adding any extra code