Invoking member function in inheritance

Function type **Invoked through**	Non-virtual	Virtual

Object variable	Class type of object variable determine which version of the function
	(static binding)
	
	One one;
	one.print(); // One::print()
	Two two;
	two.print(): //Two::print()
	one=two;
	one.print(); // One::print()
		Class type of object variable
		(static binding)
		
		One one;
		one.print(); // One::print()
		Two two;
		two.print(): //Two::print()
		one=two;
		one.print(); // One::print()

Pointer variable	Class type of pointer variable
	(static binding)
	
	One *onePtr = new One;
	onePtr->print(); // One::print()
	onePtr = new Two;
	onePtr->print(); // One::print()
		Class type of pointee
		(dynamic binding)
		
		One *onePtr = new One;
		onePtr->print(); // One::print()
		onePtr = new Two;
		onePtr->print(); // Two::print()

Passed by value
 	
TestPrint(One one){
 one.print();
}
	Class type of parameter
	(static binding)
	
	One one;
	TestPrint(one); // One::print()
	Two two;
	TestPrint(two); //One::print()
	one=two;
	TestPrint(one); //One::print()
		Class type of parameter
		(static binding)
		
		One one;
		TestPrint(one); // One::print()
		Two two;
		TestPrint(two); //One::print()
		one=two;
		TestPrint(one); //One::print()

Passed by reference
 	
TestPrint(One &one){
 one.print();
}
	Class type of parameter
	(static binding)
	
	One one;
	TestPrint(one); // One::print()
	Two two;
	TestPrint(two); // One::print()
	one=two;
	TestPrint(one); //One::print()
		Class type of parameter
		(dynamic binding)
		
		One one;
		TestPrint(one); // One::print()
		Two two;
		TestPrint(two); //Two::print()
		one=two;
		TestPrint(one); // One::print()

Pointer parameter
 	
TestPrint(One *onePtr){
 onePtr->print();
}
	Class type of parameter
	(static binding)
	
	One *onePtr = new one;
	TestPrint(onePtr); // One::print()
	
	onePtr = new Two;
	TestPrint(one); // One::print()
	
	Two *twoPtr = new Two;
	TestPrint(twoPtr); //One::print()
		Class type of pointee
		(dynamic binding)
		
		One *onePtr = new One;
		TestPrint(onePtr); // One::print()
		
		onePtr = new Two;
		TestPrint(one); // Two::print()
		
		Two *twoPtr = new Two;
		TestPrint(twoPtr); //Two::print()