


# Your Results for: " Capítulo 13"

[Print this page](#)**Book Title:** C++ como Programar**Book Author:** Deitel**Location on Site:** Exercícios de múltipla escolha (em

Inglês) &gt; Capítulo 13

**Date/Time** December 5, 2013 at 9:04 AM**Submitted:** (UTC/GMT)

## Summary of Results

**100% Correct** of 27 Scored items:27 Correct:  100%0 Incorrect:  0%[More information about scoring](#)

---

### 1. **CORRECT** Section 13.1 Introduction

13.1 Q1: Polymorphism is implemented via:

**Your Answer:** virtual functions and dynamic binding.

---

### 2. **CORRECT** Section 13.2 Base Classes and Derived Classes

13.2 Q1: Which of the following would not be a member function that derived classes Fish, Frog and Bird should inherit from base class Animal and then provide their own definitions for, so that the function call can be performed polymorphically?

**Your Answer:** lapWings.

---

### 3. **CORRECT** Section 13.3.1 Invoking Base-Class Functions from Derived-Class Objects

13.3.1 Q1: Employee is a base class and HourlyWorker is a derived class, with a redefined non-virtual print function. Given the following statements, will the output of the two print function calls be identical?

```
HourlyWorker h;  
Employee *ePtr = &h;  
  
ePtr->print();  
ePtr->Employee::print();
```

**Your Answer:** Yes.

---

### 4. **CORRECT** Section 13.3.2 Aiming Derived-Class Pointers at Base-Class Objects

13.3.2 Q1: Which of the following assignments would be a compilation error?

**Your Answer:** Assigning the address of a base-class object to a derived-class pointer.

- 
5. **CORRECT** Section 13.3.3 Derived-Class Member-Function Calls via Base-Class Pointers  
13.3.3 Q1: Downcasting enables:

**Your Answer:** Making a base-class pointer into a derived-class pointer.

---

6. **CORRECT** Section 13.3.4 Virtual Functions  
13.3.4 Q1: If objects of all the classes derived from the same base class all need to draw themselves, the draw() function would most likely be declared:

**Your Answer:** virtual.

---

7. **CORRECT** 13.3.4 Q2: Assume we have a base class Shape and derived classes Triangle and Rectangle. Which of the following member functions should be virtual?

**Your Answer:** isRegular.

---

8. **CORRECT** 13.3.4 Q3: virtual functions must:

**Your Answer:** Be declared virtual in the base class.

---

9. **CORRECT** 13.3.4 Q4: Which of the following statements about virtual functions is false?

**Your Answer:** They do not remain virtual down the inheritance hierarchy.

---

10. **CORRECT** Section 13.4 Type Fields and switch Statements  
13.4 Q1: Problems using switch logic to deal with many objects of different types do not include:

**Your Answer:** Not being able to implement separate functions on different objects.

---

11. **CORRECT** Section 13.5 Abstract Classes and Pure virtual Functions

13.5 Q1: The line:

virtual double earnings() const = 0;

---

appears in a class definition. You cannot deduce that:

**Your Answer:** All classes that directly inherit from this class will override this method.

---

**12.** **CORRECT** 13.5 Q2: Abstract classes:

**Your Answer:** Are defined, but the programmer never intends to instantiate any objects from them.

---

**13.** **CORRECT** 13.5 Q3: The main difference between a pure virtual function and a virtual function is:

**Your Answer:** That a pure virtual function cannot have an implementation.

---

**14.** **CORRECT** 13.5 Q4: Which of the following is not allowed?

**Your Answer:** Objects of abstract classes.

---

**15.** **CORRECT** Section 13.6 Case Study: Payroll System Using Polymorphism

13.6 Q1: What mistakes prevents the following class declaration from functioning properly as an abstract class?

```
class Shape
{
public:
    virtual double print() const;
    double area() const { return base * height; }
private:
    double base;
    double height;
};
```

**Your Answer:** There are no pure virtual functions.

---

**16.** **CORRECT** Section 13.7 (Optional) Polymorphism, Virtual Functions and Dynamic Binding "Under the Hood"

13.7 Q1: An abstract class will:

**Your Answer:** Have at least one 0 in its *vtable*.

---

**17.** **CORRECT** 13.7 Q2: Concrete classes that inherit virtual functions but do not override their implementations:

**Your Answer:** Receive pointers to their base classes' virtual functions.

---

**18.** **CORRECT** 13.7 Q3: Polymorphism and virtual functions are not appropriate for:

**Your Answer:** Programs that have strict memory and processor requirements.

---

**19.** **CORRECT** 13.7 Q4: The C++ compiler makes objects take up more space in memory if they:

**Your Answer:** Have virtual functions.

---

**20.** **CORRECT** 13.7 Q5: Abstract classes do not necessarily have:

**Your Answer:** Zero references to their class.

---

**21.** **CORRECT** 13.7 Q6: Which statement is not true about dynamic binding?

**Your Answer:** It eliminates the usefulness of separate header and source files.

---

**22.** **CORRECT** Section 13.8 Case Study: Payroll System Using Polymorphism and Run-Time Type Information with downcasting, `dynamic_cast`, `typeid` and `type_info`

13.8 Q1: The line:

`virtual double functionX() const = 0;` in a class definition indicates that the class is probably a:

**Your Answer:** Base class.

---

**23.** **CORRECT** 13.8 Q2: Run-time type information can be used to determine:

**Your Answer:** An object's type.

---

**24.** **CORRECT** 13.8 Q3: The \_\_\_\_\_ operator returns a reference to a \_\_\_\_\_ object:

**Your Answer:** typeid, type\_info.

---

**25.** **CORRECT** 13.8 Q4: dynamic\_cast is often used to:

**Your Answer:** Downcast pointers.

---

**26.** **CORRECT** Section 13.9 Virtual Destructors

13.9 Q1: virtual destructors must be used when:

**Your Answer:** delete is used on a base-class pointer to a derived-class object.

---

**27.** **CORRECT** Section 13.10 (Optional) Software Engineering Case Study: Incorporating Inheritance into the ATM System

13.10 Q1: Which attribute or behavior would we not factor out of the Pants and Socks classes and into the Clothing base class?

**Your Answer:** numberOfPockets.

---

### E-mail Your Results

My name is (first last):

E-mail my results to:

**E-mail address:**

**Send as:**

☐ Me

Text

☐ Instructor

Text

☐ TA

Text

☐ Other

Text

[Help](#)



Copyright © 1995 - 2010 [Pearson Education](#). All rights reserved.  
[Legal Notice](#) | [Privacy Policy](#) | [Permissions](#)