Peer Instruction 3

Polymorphism
Which one of the following instantiations (with typecast) is incorrect?

A. Employee e = new Person();
B. Employee e = new Employee();
C. Person p = new Employee();
D. Person p = new HourlyEmployee();
E. Person p = new SalariedEmployee();

Incorrect answer is A, cannot cast the superclass to subclass reference, Person is not an instance of Employee.
Which is the incorrect statement about implicit or explicit constructor chaining?

Please select the single incorrect answer.

A. Implicit chaining is when the compiler inserts default constructors.
B. Explicit chaining requires the programmer to insert constructors.
C. When chaining, the super class constructor must be called first.
D. The programmer can explicitly chain default constructors.
E. Implicit chaining applies when there is no default constructor.

Incorrect answer is E, no implicit chaining possible when there is not default constructor.
Which version of printPaycheck is called when the following code is executed?

```java
Employee e = new HourlyEmployee();
e.printPaycheck();
```

A. printPaycheck in Person  
B. printPaycheck in Employee  
C. printPaycheck in HourlyEmployee  
D. printPaycheck in SalariedEmployee  
E. Will not compile

Correct answer is C), this is polymorphism as implemented through dynamic binding.
Which version of printPaycheck is called when the following code is executed?

Person p = new Employee();
p.printPaycheck();

A. printPaycheck in Person
B. printPaycheck in Employee
C. printPaycheck in HourlyEmployee
D. printPaycheck in SalariedEmployee
E. Will not compile

Correct answer is E, Person class has no printPaycheck, so it is filtered even though Employee object has it!
Which statement is correct about overriding private methods in the super class?

Please select the single correct answer.
A. Any derived class can override private methods in the base class.
B. Private methods can never be overridden by a derived class.
C. Any derived class in the same package can override private methods.
D. Overriding a private method hides the base class version.
E. Overriding private methods is allowed but highly discouraged.

Correct answer is B, cannot override private methods, they are private to the base class (and final).
What happens when a derived class overrides a public method with a private version?

Please select the single correct answer.

A. Method remains public in the base class but is private in the derived.
B. Compiles but a runtime error occurs when calling the method.
C. Does not compile because the derived cannot increase restriction.
D. Immediately makes the method private in the base class as well.
E. Hard to say what this might do!

Correct answer is C, cannot increase the restriction, must be equal to or more accessible.
Which is printed when the following code with getClass calls is executed?

```java
Person p = new Employee();
Employee e = new Employee();
System.out.println(p.getClass()+","+e.getClass());
```

A. class Person,class Employee  
B. class Employee,class Employee  
C. class Person,class Person  
D. class Employee,class Person  

Correct answer is B, objects are instances of what they were created as, and getClass returns the same.  
Note: Can often use getClass and instanceof in an equivalent manner.
Which of the following is an incorrect statement about access modifiers?

Please select the single incorrect answer.

A. Public means methods and attributes accessible from anywhere.
B. Private means accessible only from within the class where defined.
C. Protected has a different meaning for methods and attributes.
D. Protected means accessible from same package or any subclass.
E. Default means no other access modifier has been specified.

Incorrect answer is C, access modifiers are identical for methods and attributes. Note: Default differs very slightly from protected.
Which statement about the final keyword is incorrect?

Please select the single incorrect answer.

A. When final is applied to a class variable, it cannot be modified.
B. When final is applied to a local variable, it cannot be modified.
C. When final is applied to a parameter, it cannot be modified.
D. When final is applied to a method, it cannot be overridden.
E. When final is applied to a class, it can be extended but not instantiated.

Incorrect answer is E, final prevents the class from being extended (inherited), still can be instantiated.
Question: Can all classes be instantiated? After all that’s what they’re for. Answer: Not if they are abstract.