1. What will be the result of compiling and running the following program?

public class Polymorphism {

public static void main(String[] args) {

A ref1 = new C();

B ref2 = (B) ref1;

System.out.println(ref2.f());

}

}

class A { int f() { return 0; } }

class B extends A { int f() { return 1; } }

class C extends B { int f() { return 2; } }

Select the one correct answer.

(a) The program will fail to compile.

(b) The program will compile, but will throw a ClassCastException at

runtime.

(c) The program will compile, and print 0 when run.

(d) The program will compile, and print 1 when run.

(e) The program will compile, and print 2 when run.

*(e)*

The program will print 2 when System.out.println(ref2.f()) is executed. The object referenced by ref2 is of class C, but the reference is of type B. Since B contains a method f(), the method call will be allowed at compile time.

During execution it is determined that the object is of class C, and dynamic method lookup will cause the overriding method in C to be executed.

1. What will be the result of compiling and running the following program?

public class Polymorphism2 {

public static void main(String[] args) {

A ref1 = new C();

B ref2 = (B) ref1;

System.out.println(ref2.g());

}

}

class A {

private int f() { return 0; }

public int g() { return 3; }

}

class B extends A {

private int f() { return 1; }

public int g() { return f(); }

}

class C extends B {

public int f() { return 2; }

}

Select the one correct answer.

(a) The program will fail to compile.

(b) The program will compile, and print 0 when run.

(c) The program will compile, and print 1 when run.

(d) The program will compile, and print 2 when run.

(e) The program will compile, and print 3 when run.

*(c)*

The program will print 1 when run. The f() methods in A and B are private,

and are not accessible by the subclasses. Because of this, the subclasses cannot

overload or override these methods, but simply define new methods with the same signature. The object being called is of class C. The reference used to access the object is of type B. Since B contains a method g(), the method call will be allowed at compile time. During execution it is determined that the object is of class C, and dynamic method lookup will cause the overriding method g() in B to be executed.

This method calls a method named f. It can be determined during compilation that this can refer to only the f() method in B, since the method is private and

cannot be overridden. This method returns the value 1, which is printed.

1. What is the return type of the hashCode() method in the Object class?

Select the one correct answer.

(a) String

(b) int

(c) long

(d) Object

(e) Class

*(b)*

The method hashCode() in the Object class returns a hash code value of type int.

1. Which of the following statements is true?

Select the one correct answer.

(a) If the references x and y denote two different objects, the expression x.equals(y) is always false.

(b) If the references x and y denote two different objects, the expression (x.hashCode() == y.hashCode()) is always false.

(c) The hashCode() method in the Object class is declared as final.

(d) The equals() method in the Object class is declared as final.

(e) All arrays have a method named clone.

*(e)*

All arrays are genuine objects and inherit all the methods defined in the Object class, including the clone() method. Neither the hashCode() method nor the equals() method is declared as final in the Object class, and it cannot be guaranteed that implementations of these methods will differentiate among all objects.

1. Which exception can the clone() method of the Object class throw?

Select the one correct answer.

(a) CloneNotSupportedException

(b) NotCloneableException

(c) IllegalCloneException

(d) NoClonesAllowedException

*(a)*

The clone() method of the Object class will throw a

CloneNotSupportedException if the class of the object does not implement the Cloneable interface.

1. Which of the following are wrapper classes?

Select the three correct answers.

(a) java.lang.Void

(b) java.lang.Int

(c) java.lang.Boolean

(d) java.lang.Long

(e) java.lang.String

*(a), (c), and (d)*

The class java.lang.Void is considered a wrapper class, although it does not wrap any value. There is no class named java.lang.Int, but there is a wrapper class named java.lang.Integer. A class named java.lang.String also exists, but it is not a wrapper class since all strings in Java are objects.

1. Which of the following classes do not extend the java.lang.Number class?

Select the two correct answers.

(a) java.lang.Float

(b) java.lang.Byte

(c) java.lang.Character

(d) java.lang.Boolean

(e) java.lang.Short

*(c) and (d)*

The classes Character and Boolean are non-numeric wrapper classes and do not extend the Number class. The classes Byte, Short, Integer, Long, Float, and Double are numeric wrapper classes that extend the abstract Number class.

1. Which of these classes define immutable objects?

Select the three correct answers.

(a) Character

(b) Byte

(c) Number

(d) Short

(e) Object

*(a), (b), and (d)*

All instances of concrete wrapper classes are immutable. The Number class is an abstract class.

1. Which of these classes have a single-parameter constructor taking a string?

Select the two correct answers.

(a) Void

(b) Integer

(c) Boolean

(d) Character

(e) Object

*(b) and (c)*

All instances of wrapper classes except Void and Character have a constructor that accepts a single String parameter. The class Object has only a no-argument constructor.

1. Which of the wrapper classes have a booleanValue() method?

Select the one correct answer

(a) All wrapper classes

(b) All wrapper classes except Void

(c) All wrapper classes that also implement the compareTo() method

(d) All wrapper classes extending Number

(e) Only the class Boolean

*(e)*

While all numeric wrapper classes have the methods byteValue(),

doubleValue(), floatValue(), intValue(), longValue(), and

shortValue(), only the Boolean class has the booleanValue() method. Likewise, only the Character class has the charValue() method.

1. hich statements are true about wrapper classes?

Select the two correct answers.

(a) String is a wrapper class.

(b) Double has a compareTo() method.

(c) Character has a intValue() method.

(d) Byte extends Number.

*(b) and (d)*

String is not a wrapper class. All wrapper classes except Void have a compareTo() method. Only the numeric wrapper classes have an intValue() method. The Byte class, like all other numeric wrapper classes, extends the Number class.

1. What will the following program print when compiled and run?

public class RQ200A70 {

public static void main(String[] args) {

Integer i = new Integer(-10);

Integer j = new Integer(-10);

Integer k = -10;

System.out.print((i==j) + “|”);

System.out.print(i.equals(j) + “|”);

System.out.print((i==k) + “|”);

System.out.print(i.equals(k));

}

}

Select the one correct answer.

(a) false|true|false|true

(b) true|true|true|true

(c) false|true|true|true

(d) true|true|false|true

(e) None of the above.

*(a)*

Using the new operator creates a new object. Boxing also creates a new object if one is not already interned from before.

1. Which of the following operators cannot have an operand of type String?

Select the two correct answers.

(a) +

(b) -

(c) +=

(d) .

(e) &

*(b) and (e)*

The operators - and & cannot be used in conjunction with a String object. The operators + and += perform concatenation on strings, and the dot operator accesses members of the String object.

1. Which of the following statements are true?

Select the two correct answers.

(a) String objects are immutable.

(b) Subclasses of the String class can be mutable.

(c) All wrapper classes are declared as final.

(d) All objects have a public method named clone.

(e) The expression ((new char[] {'o', 'k'}) instanceof

String) is always true.

*(a) and (c)*

The String class and all wrapper classes are declared as final and, therefore, cannot be extended. The clone() method is declared as protected in the Object class. String objects and wrapper class objects are immutable and, therefore, cannot be modified. The class String and char array types are unrelated, resulting in a compile-time error.