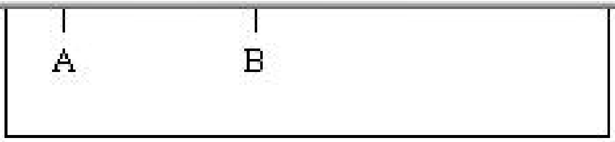
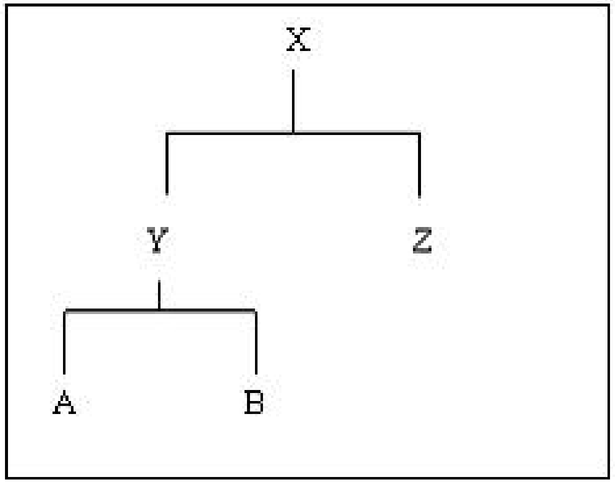


A is a derived class of X.

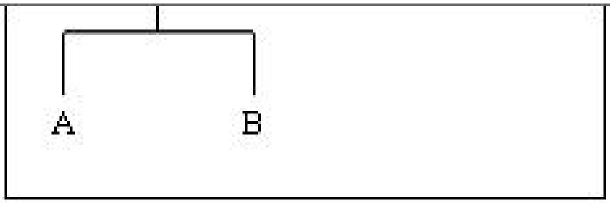
Answer: F
While A is a descendant of class X, it is not a derived class. Instead, X is a derived class of Y and Y is a derived class of X. A ultimately inherits items from X as long as those items were not overridden by Y.
Points: 0 / 1

❌___ 2. Consider the following class hierarchy and answer question



Y is a derived class of Z.

Answer: F
Y and Z are known as siblings because they are children of the same parent, but Y and Z may have nothing in common other than what they both inherit from X
Points: 0 / 1

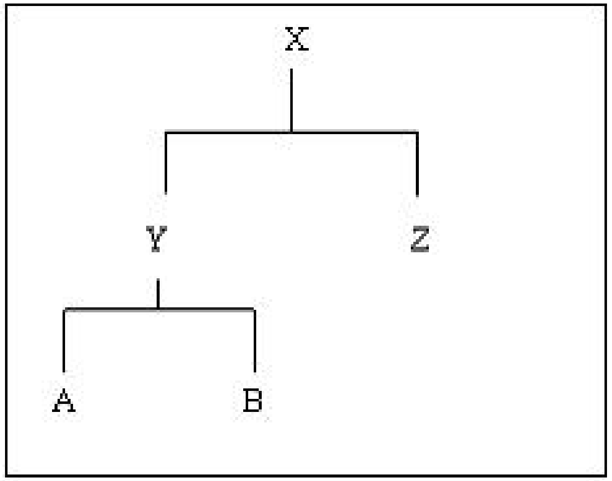


If A, B, Y and Z all contain the same instance data d1, then d1 should be declared in X and inherited into Y, Z, A and B.

Answer: T
Anything in common between A, B, Y and Z should be declared in X to properly identify that it is a class trait common amongst them. That is, common elements should be defined as high up in the hierarchy as possible. If all descendants of X have the trait (instance data) d1, then it is part of X too

Points: 0 / 1

4. Consider the following class hierarchy and answer question



B inherits any public methods defined in X (that are not overridden by Y)

Answer: T

✗ ___ 5. A derived class has access to all of the methods of the parent class, but not to any of the instance data of the parent class.

Answer: F
Since methods can be declared as private, any private methods are not accessible by a derived class. Furthermore, any public (or protected) instance variables are accessible in a derived class
Points: 0 / 1

✗ ___ 6. If class AParentClass has a public instance data x, and AChildClass is a derived class of AParentClass, then AChildClass can access x but can not redefine x to be a different type.

Answer: F
A derived class can redefine any of the instance data or methods of the parent class. The parent class' version is now hidden, but can be accessed through the use of super, as in super.x
Points: 0 / 1

✗ ___ 7. Assume that Poodle and Retriever are derived classes of Dog and that Dog d = new Dog(…), Poodle p = new Poodle(…), and Retriever r = new Retriever(…); where the … are the necessary parameters for the classes.

The assignment statement d = p; is legal even though d is not a Poodle.

Answer: T
Since Poodle extends Dog, Dog is a wider class (Poodle is more specific, and therefore narrower). An assignment statement can assign a narrower item into a wider variable (since p is narrower, it can be assigned to d)
Points: 0 / 1

✗ ___ 8. Assume that Poodle and Retriever are derived classes of Dog and that Dog d = new Dog(…), Poodle p = new Poodle(…), and Retriever r = new Retriever(…); where the … are the necessary parameters for the classes.

The assignment statement p = d; is legal even though p is not a Dog.

Answer:	F Since Poodle extends Dog, Dog is known as a wider class (Poodle is more specific, and therefore narrower). In any assignment statement, it is possible to take a narrower item and assign it into a wider variable, but it is not possible to assign a wider item into a narrower variable. To accomplish this, a cast must be explicitly made. So, the statement p = d; is illegal although the statement p = (Poodle) d; would be legal
Points:	0 / 1

✗ ____ 9. Assume that Poodle and Retriever are derived classes of Dog and that Dog d = new Dog(…), Poodle p = new Poodle(…), and Retriever r = new Retriever(…); where the … are the necessary parameters for the classes.

The statements d = p; d = r; demonstrate polymorphism.

Answer: T
A polymorphic reference is one that can refer to different types of objects at different times. The reference d refers to a Poodle after the first assignment, and to a Retriever after the second assignment
Points: 0 / 1

✗ ____ 10. Inheritance allows us to create a new class from an existing class

Answer: T
This is a powerful technique that allows us to reuse software
Points: 0 / 1

✗ ____ 11. The word super is used to refer to the parent class.

Answer: T
super can be used to call the constructor and methods of the parent class.
Points: 0 / 1

✗ ____ 12. In Java, each child class can have only on parent.

Answer: T
This is called single inheritance and it is the approach used in Java (as opposed to multiple inheritance)
Points: 0 / 1

multiple inheritance)

Points: 0 / 1

✗ ___ 13. Polymorphism works with inheritance only; it does not work with interfaces

Answer: F
Polymorphism occurs with interfaces as well

Points: 0 / 1

✗ ___ 14. Every object in Java, no matter what class it came from, has a method called toString

Answer: T
The toString method is on the Object class, which every other object in Java descends from

Points: 0 / 1

✗ ____ 15. When using inheritance, the relationship between the child and the parent should be a has-a relationship.

Answer: F
False. Explanation: The relationship should be an is-a relationship. The is the child *is-a* parent. For example, a Poodle is-a Dog.
Points: 0 / 1

✗ ____ 16. When a child defines a method with the same signature as a method in the parent, this is called method overloading.

Answer: F
False. Explanation: Method overloading is two methods in the same class that have the same name but different parameters. When a child defines a method with the same signature as a method in the parent, this is called method overriding.
Points: 0 / 1

MULTIPLE CHOICE

✗ ____ 17. Java does not support multiple inheritance, but some of the abilities of multiple inheritance are available by

a. importing classes
b. implementing interfaces
c. overriding parent class methods
d. creating alias
e. using public rather than protected or private modifiers

Answer: B
Since a class can implement any number of interfaces, that class is in essence using

✗ ___ 18. Which of the following is not a method of the Object class?

a. hashCode
b. compareTo
c. equals
d. toString
e. all of the above are methods of the Object clas

Answer: B
The Object class defines clone to create a copy of any object, equals to determine if two objects are the same object, and toString to translate an Object into a String. However, compareTo is not implement by Object and must be explicitly implemented in any class that wants to implement the Comparable interface.
Points: 0 / 1

Name: Jeff Wallace **ID: 1018**

✗ ___ 19. All classes in Java are directly or indirectly subclasses of the _______ class.

a. Wrapper
b. String
c. Reference
d. this
e. Object

Answer: E
Java requires that all classes having a parent class. If a class does not extend another

20

Consider that you want to extend AClass to BClass. BClass will have a third int instance data, z. Which of the following would best define BClass' constructor?

a.
```
public BClass(int a, int b, int c)
{
super(a, b, c);
}
```

b.
```
public BClass(int a, int b, int c)
{
x = a;
y=b;
z=c;
}
```

c.
```
public BClass(int a, int b, int c)
{
z=c;
}
```

d.
```
public BClass(int a, int b, int c)
{
super(a, b);
z = c;
}
```

e.
```
public BClass(int a, int b, int c)
{
super();
}
```

Answer: D

Inheritance is useful because it allows you to reuse code. For this problem, you

21

You want addEm to now add all three values and return the sum and changeEm to change x and y, but leave z alone. Which should you do?

a. Redefine addEm and changeEm without referencing super.addEm( ) or super.changeEm( )

b. Redefine addEm to return the value of z + super.addEm( ), but leave changeEm alone

c. Redefine changeEm to call super.changeEm( ) and then set z = x + y, but leave addEm alone

d. Redefine addEm to return the value of z + super.addEm( ) and redefine changeEm to call super.changeEm( ) and then set z = x + y

e. Redefine changeEm to call super.changeEm( ) without doing anything to z, and redefine addEm to return super.addEm( )

Answer: B

Since super.changeEm( ) will not affect z and we don't want changeEm to affect z, there is no need to redefine changeEm for BClass. Any reference to changeEm( ) will reference AClass' version which is the same one that BClass will use. But addEm needs redefining because the inherited method addEm calls AClass' addEm which only adds x and y together. Now, we want to add x, y and z together. We can either redefine addEm entirely, or use super.addEm( ) and add z to the value

22

Which of the following would best redefine the toString method for BClass?

a.
```
public String toString(int z)
{
return “ ” + x + “   ” + y + “   ” + z;
}
```

b.
```
public String toString( )
{
return super.toString( )
}
```

c.
```
public String toString( )
{
return super.toString( ) + “   ” + z;
}
```

d.
```
public String toString( )
{
return super.toString( ) + “   ” x + “
” + y + “   ” + z;
}
```

e.
```
public String toString( )
{
return “ ” + x +    ” + y + “   ” + z;
}
```

Answer: C

The best way to redefine a method is to reuse the parent class’ method and supplement it with the code needed to handle the changes in the subclass. Answer c does this. Answer b returns only x and y, not z, while answer a is syntactically invalid because the toString method does not accept a parameter.

Points: 0 / 1

✗ ___ 23. Two children of the same parent class are known as

a. aliases
b. relatives
c. clones
d. brothers
e. siblings

Answer: E

The relationship between two children of the same parent is referred to as siblings (brothers would imply a gender). Clones are copies of the same object and aliases are the same object. Brothers and relatives are not used to define relationships between classes in Java.

Points: 0 / 1

✗ ___ 24. In order to determine the type that a polymorphic variable refers to, the decision is made

a. by the programmer at the time the program is written
b. by the compiler at compile time
c. by the operating system when the program is loaded into memory
d. by the Java run-time environment at run time
e. by the user at run time

Answer: D

The polymorphic variable can take on many different types, but it is not know which type it has taken on until the program is executing. At the time the variable is referenced, then the decision must be made. That decision is made by the run-time environment based on the latest assignment of the variable

Points: 0 / 1

____ 25. Use the following partial class definitions:

```
public class A1
                  {
                              public int x ;
                              private int y;
                              public int z;
                  ...
                  }

                  public class A2 extends A1
                  {
                              public int a;
                              private int b;
                  ...
                  }

                  public class A3 extends A2
                  {
                              private int q;
                  ...
                  }
```

Which of the following is true with respect to A1, A2 and A3?

a. A1 is a subclass of A2 and A2 is a subclass of A3

b. A3 is a subclass of A2 and A2 is a subclass of A1

c. A1 and A2 are both subclasses of A3

d. A2 and A3 are both subclasses of A1

e. A1, A2 and A3 are all subclasses of the class A

Answer: B

The reserved word extends is used to create a subclass. Since A2 extends A1, A2 is

✗ ____ 26. Use the following partial class definitions:

```
public class A1
        {
                public int x;
                private int y;
                public int z;
        …
        }

        public class A2 extends A1
        {
                public int a;
                private int b;
        …
        }

        public class A3 extends A2
        {
                private int q;
        …
        }
```

Which of the following lists of instance data are accessible in class A2?

a. x, y, z, a, b	d. z, a, b
b. x, y, z, a	e. a, b
c. x, z, a, b	

Answer: C
Class A2 has access to all of its own instance data (a, b) and any instance data from A1 that were protected (z). Further, all classes, including A2, have access to A1's x since it is public, so x, z, a and b are accessible in A2

Points: 0 / 1

___ 27. Use the following partial class definitions:

```
public class A1
                {
                        public int x ;
                        private int y;
                        public int z;
                ...
                }

                public class A2 extends A1
                {
                        public int a;
                        private int b;
                ...
                }

                public class A3 extends A2
                {
                        private int q;
                ...
                }
```

Which of the following lists of instance data are accessible in A3?

a. x, y, z, a, b, q

b. a, b, q

c. a, q

d. x, z, a, q

e. x, a, q

Answer: D

Obviously q is accessible in A3. Also, A3 has access to anything inherited from A2, which is a. Additionally, since A2 inherits z from A1, it is also inherited by A3. Finally, since x is public, it is visible to all classes

✗ ____ 28. Use the following partial class definitions:

```
public class A1
                {
                        public int x;
                        private int y;
                        public int z;
                …
                }

                public class A2 extends A1
                {
                        public int a;
                        private int b;
                …
                }

                public class A3 extends A2
                {
                        private int q;
                …
                }
```

Which of the following is true regarding the use of instance data y of class A1?

a. it is accessible in A1, A2 and A3
b. it is accessible in A1 and A2
c. it is accessible only in A1
d. it is accessible only in A3
e. it is not accessible to any of the three classes

Answer: C
Because it is declared as a private instance data, it is not inherited by any subclasses, and therefore y is only accessible in the class it was defined, A1

Points: 0 / 1

❌ ____ 29. Which of the following is true regarding Java classes?

a. All classes must have 1 parent but may have any number of children (derived or extended) classes

b. All classes must have 1 child (derived or extended) class but may have any number of parent classes

c. All classes must have 1 parent class and may have a single child (derived or extended) class

d. All classes can have any number (0 or more) of parent classes and any number of children (derived or extended) classes

e. All classes can have either 0 or 1 parent class and any number of children (derived or extended) classes

Answer: A
Java supports inheritance but not multiple inheritance, so a Java class can have any number of children but only one parent. Further, since all Java classes inherit either directly or indirectly from the Object class, all Java classes have exactly 1 parent class

Points: 0 / 1

❌ ____ 30. 1) A variable declared to be of one class can later reference an extended class of that class. This variable is known as

a. public

b. derivable

c. cloneable

d. polymorphic

e. none of the above, a variable declared to be of one class can never reference any other type of class, even an extended class

Answer: D
The term polymorphic means that the variable can have many forms. Under ordinary

✗ ____ 31.

The instruction super( ); does which of the following?

a. calls the method super as defined in the current class
b. calls the method super as defined in the current class' parent class
c. calls the method super as defined in java.lang
d. calls the constructor as defined in the current class
e. calls the constructor as defined in the current class' parent cl

Answer: E

The instruction super represents an invocation of something in the current class' parent class. Since there is no message but merely super( ), it is an invocation of the parent class' constructor

Points: 0 / 1

✗ ____ 32. Which of the following is an example of multiple inheritance?

a. a computer can be a mainframe or a PC
b. a PC can be a desktop or a lapt
c. a laptop is both a PC and a portable device
d. a portable device is a lightweight de
e. Macintosh and IBM PC are both types of PCs

Answer: C

c. Explanation: Multiple inheritance means that a given class inherits from more than one parent class. Of those listed above, a laptop computer inherits properties from both PCs and portable devices. The answers in a, b and e are all examples of single inheritance where a class has at least two children (in a, computer has children mainframe and PC, in b, PC has children desktop and laptop, in e, PC has children Macintosh and IBM PC). Answer d denotes a property of a class.

Points: 0 / 1