**Object Test Your Understanding**

1: If you don’t specify the parent class in a class declaration which of the following is true?

Top of Form

(A) It doesn't have a parent class.  
(B) It inherits from the Object class.  
(C) It inherits from the Default class.  
(D) It inherits from the Parent class.

2: If the class Vehicle has object fields of make and model and the class Car inherits from the class vehicle will a car object have a make and model?

Top of Form

(A) Yes  
(B) No

Bottom of Form

3: If I had a class ParkingGarage should it inherit from the class Vehicle?

Top of Form

(A) Yes  
(B) No  
Bottom of Form

4: In Java how many parents can a class have?

Top of Form

(A) 0  
(B) 1  
(C) 2  
(D) infinite  
Bottom of Form

**2 .Check your understanding**

2-1: A bookstore is working on an on-line ordering system. For each type of published material (books and movies) they need to track the id, title, author(s), date published, and price. Which of the following would be the best design?

Top of Form

(A) Create one class PublishedMaterial with the requested fields plus type  
(B) Create classes Book and Movie and each class has the requested fields  
(C) Create the class PublishedMaterial and have Book and Movie inherit from it all the listed fields  
(D) Create one class BookStore with the requested fields plus type  
(E) Create classes for PublishedMaterial, Books, Movies, Title, Price, ID, Authors, DatePublished

2-2: A movie theater has multiple showings of a movie each day. Each movie showing has a start time and location (theater number). What should the relationship be between the Movie class and the MovieShowing class?

Top of Form

(A) The MovieShowing class should be a subclass of the Movie class.  
(B) The Movie class should be a subclass of the MovieShowing class.  
(C) A MovieShowing has a movie associated with it, so it should have a Movie field.  
Bottom of Form

2-3: What Java keyword is used to specify the parent class?

Top of Form

(A) superclass  
(B) parent  
(C) extends  
(D) classBottom of Form

**2-4: Question: Which of the following reasons for using an inheritance heirarchy are valid?**

1. Object methods from a superclass can be used in a subclass without rewriting or copying code.
2. Objects from subclasses can be passed as arguments to a method that takes an argument of the parent type.
3. Objects from subclasses can be stored in the same array of the parent type.
4. All of the above
5. None of the above

Top of Form

(A) V  
(B) IV  
(C) I and II  
(D) I and III  
(E) I onlyBottom of Form

**3. Check your understanding**

3-1: Which of the following declarations in Student would correctly *override* the getFood method in Person?

**public** **class** **Person**

{

**private** String name = **null**;

**public** Person(String theName)

{

name = theName;

}

**public** String getFood()

{

**return** "Hamburger";

}

}

**public** **class** **Student** **extends** Person

{

**private** int id;

**private** **static** int nextId = 0;

**public** Student(String theName)

{

**super**(theName);

id = nextId;

nextId++;

}

**public** int getId() {**return** id;}

**public** void setId (int theId)

{

**this**.id = theId;

}

}

1. Top of Form

(A) public void getFood()  
(B) public String getFood(int quantity)  
(C) public String getFood()

1. Bottom of Form

3-2: Which of the following declarations in Person would correctly *overload* the getFood method in Person?

**public** **class** **Person**

{

**private** String name = **null**;

**public** Person(String theName)

{

name = theName;

}

**public** String getFood()

{

**return** "Hamburger";

}

}

**public** **class** **Student** **extends** Person

{

**private** int id;

**private** **static** int nextId = 0;

**public** Student(String theName)

{

**super**(theName);

id = nextId;

nextId++;

}

**public** int getId() {**return** id;}

**public** void setId (int theId)

{

**this**.id = theId;

}

}

1. Top of Form

(A) public void getFood()  
(B) public String getFood(int quantity)  
(C) public String getFood()

1. Bottom of Form

4. **Check your understanding**

4-1: Given the following class declarations, and assuming that the following declaration appears in a client program: Base b = new Derived();, what is the result of the call b.methodOne();?

**public** **class** **Base**

{

**public** void methodOne()

{

System.out.print("A");

methodTwo();

}

**public** void methodTwo()

{

System.out.print("B");

}

}

**public** **class** **Derived** **extends** Base

{

**public** void methodOne()

{

**super**.methodOne();

System.out.print("C");

}

**public** void methodTwo()

{

**super**.methodTwo();

System.out.print("D");

}

}

**5.Check your understanding**

5-1: Given the following class definitions which of the following would not compile if it was used in place of the missing code in the main method?

**class** **Item**

{

**private** int x;

**public** void setX(int theX) { x = theX; }

*// ... other methods not shown*

}

**public** **class** **EnhancedItem** **extends** Item

{

**private** int y;

**public** void setY(int theY) { y = theY; }

*// ... other methods not shown*

**public** **static** void main(String[] args)

{

EnhancedItem currItem = **new** EnhancedItem();

*// missing code*

}

}

(A) currItem.setX(3);  
(B) currItem.setY(2);  
(C) currItem.x = 3;  
(D) currItem.y = 2;

1. **Check your understanding**

6-1: Given the class definitions of Point2D and Point3D below, which of the constructors that follow (labeled I, II, and III) would be valid in the Point3D class?

**class** **Point2D** {

**public** int x;

**public** int y;

**public** Point2D() {}

**public** Point2D(int x,int y) {

**this**.x = x;

**this**.y = y;

}

*// other methods*

}

**public** **class** **Point3D** **extends** Point2D

{

**public** int z;

*// other code*

}

*// possible constructors for Point3D*

I. **public** Point3D() {}

II. **public** Point3D(int x, int y, int z)

{

**super**(x,y);

**this**.z = z;

}

III. **public** Point3D(int x, int y)

{

**this**.x = x;

**this**.y = y;

**this**.z = 0;

}

6-2: Given the class definitions of Point and NamedPoint below, which of the constructors that follow (labeled I, II, and III) would be valid in the NamedPoint class?

**class** **MPoint**

{

**private** int myX; *// coordinates*

**private** int myY;

**public** MPoint( )

{

myX = 0;

myY = 0;

}

**public** MPoint(int a, int b)

{

myX = a;

myY = b;

}

*// ... other methods not shown*

}

**public** **class** **NamedPoint** **extends** MPoint

{

**private** String myName;

*// constructors go here*

*// ... other methods not shown*

}

*// Proposed constructors for this class:*

I. **public** NamedPoint()

{

myName = "";

}

II. **public** NamedPoint(int d1, int d2, String name)

{

myX = d1;

myY = d2;

myName = name;

}

III. **public** NamedPoint(int d1, int d2, String name)

{

**super**(d1, d2);

myName = name;

}

(A) I only  
(B) I and III  
(C) II only  
(D) III only

**8. Check your understanding**

8-1: Which of the following is true about interfaces?

I. Interfaces can only contain **abstract** methods or **class** **constants**.

II. Interfaces can be extended.

III. Interfaces can be instantiated (you can create an object of the **interface** **type**).

Top of Form

(A) I only  
(B) II only  
(C) I and II  
(D) I, II, and III

**9. Check your understanding**

10-9-2: What is the output from running the main method in the GradStudent class?

**public** **class** **Student** {

**public** String getFood() {

**return** "Pizza";

}

**public** String getInfo() {

**return** **this**.getFood();

}

**public** **static** void main(String[] args)

{

Student s1 = **new** GradStudent();

s1.getInfo();

}

}

**class** **GradStudent** **extends** Student {

**public** String getFood() {

**return** "Taco";

}

}

Top of Form

(A) Pizza  
(B) Taco  
(C) You will get a compile time error  
(D) You will get a run-time error  
Check MeCompare me

Bottom of Form

10-9-3: What is the output from running the main method in the RaceCar class?

**public** **class** **Car**

{

**private** int fuel;

**public** Car() { fuel = 0; }

**public** Car(int g) { fuel = g; }

**public** void addFuel() { fuel++; }

**public** void display() { System.out.print(fuel + " "); }

**public** **static** void main(String[] args)

{

Car car = **new** Car(5);

Car fastCar = **new** RaceCar(5);

car.display();

car.addFuel();

car.display();

fastCar.display();

fastCar.addFuel();

fastCar.display();

}

}

**class** **RaceCar** **extends** Car

{

**public** RaceCar(int g) { **super**(2\*g); }

}

Top of Form

(A) 5 6 10 11  
(B) 5 6 5 6  
(C) 10 11 10 11  
(D) The code won't compile.

9-4: Given the following class definitions and a declaration of Book b = new Dictionary which of the following will cause a compile-time error?

**public** **class** **Book**

{

**public** String getISBN()

{

*// implementation not shown*

}

*// constructors, fields, and other methods not shown*

}

**public** **class** **Dictionary** **extends** Book

{

**public** String getDefinition(String word)

{

*// implementation not shown*

}

}

Top of Form

(A) b.getISBN();  
(B) b.getDefintion();  
(C) ((Dictionary) b).getDefinition();

Bottom of Form

9-5: Assume that the following declaration appears in a client program **Base b = new Derived();**. What is the result of the call **b.methodOne()**?

**public** **class** **Base**

{

**public** void methodOne()

{

System.out.print("A");

methodTwo();

}

**public** void methodTwo()

{

System.out.print("B");

}

**public** **static** void main(String[] args)

{

Base b = **new** Derived();

b.methodOne();

}

}

**class** **Derived** **extends** Base

{

**public** void methodOne()

{

**super**.methodOne();

System.out.print("C");

}

**public** void methodTwo()

{

**super**.methodTwo();

System.out.print("D");

}

}

Top of Form

(A) ABDC  
(B) AB  
(C) ABCD  
(D) ABC