Quiz 08

Due Oct 28 at 12pm **Points** 10 **Questions** 10 **Time Limit** None

Instructions

There is no time limit, but you may only make one submission.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	18 minutes	10 out of 10

Score for this quiz: **10** out of 10

Submitted Oct 22 at 4:57pm

This attempt took 18 minutes.

Question 1

1 / 1 pts

Consider the following code snippet:

```
public class Vehicle
{
    . . .
    public void setVehicleAttributes()
    {
        . . .
    }
}

public class Auto extends Vehicle
{
    . . .
    public void setVehicleAttributes()
    {
        . . .
    }
}
```

Which of the following statements is correct?

Correct!

◉ The subclass is overriding a superclass method.

○ This code will not compile.

○ The subclass is overloading a superclass method.

○ The subclass is shadowing a superclass method.

Question 2

1 / 1 pts

You are creating a `Motorcycle` class which is supposed to be a subclass of the `Vehicle` class. Which of the following class declaration statements will accomplish this?

Correct!

● `public class Motorcycle extends Vehicle`

○ `public class Motorcycle inherits Vehicle`

○ `public class Motorcycle implements Vehicle`

○ `public class Motorcycle interfaces Vehicle`

Question 3

1 / 1 pts

Consider the following code snippet:

```
Vehicle aVehicle = new Auto();
aVehicle.moveForward(200);
```

If the `Auto` class inherits from the `Vehicle` class, and both classes have an implementation of the `moveForward()` method with the same set of parameters and the same return type, which statement is correct?

Correct!

● The `moveForward()` method of the `Auto` class will be executed.

○ You must specify in the code which class's `moveForward()` method is to be used.

○ It is not possible to determine which class's method is called.

○ The `moveForward()` method of the `Vehicle` class will be executed.

Question 4

1 / 1 pts

Suppose the class `Value` is partially defined below

```
public class Value
{
    private int number;

    public int getValue()
    {
        return number;
    }
}
```

A subclass of `Value`, `LargerValue`, is defined with a `getValue` method that returns twice the value of the parent. Which line is the body of `LargerValue`'s `getValue` method?

○ `return getValue() * 2;`

○ `return number * 2;`

○ `return super.number * 2;`

Correct!

● `return super.getValue() * 2;`

Question 5

1 / 1 pts

You are creating a class inheritance hierarchy about motor vehicles that will contain classes named `Vehicle`, `Auto`, and `Motorcycle`. Which of the following statements is correct?

○ `Vehicle` should be the subclass, while `Auto` and `Motorcycle` should be the superclasses.

Correct!

● `Vehicle` should be the superclass, while `Auto` and `Motorcycle` should be the subclasses.

○ `Vehicle` should be the default class, while `Auto` and `Motorcycle` should be the subclasses.

○ `Vehicle` should be the subclass, while `Auto` and `Motorcycle` should be the default classes.

Question 6

1 / 1 pts

A very simple, limited-use class can be declared entirely within another class (or within one of its methods). What is the very simple class of this arrangement called?

○ method

○ callback

○ outer class

Correct!

● inner class

Question 7

1 / 1 pts

Consider the following code snippet.

```
public interface Measurable
{
    double getMeasure();
}
```

```
public class Coin implements Measurable
{
    public Coin(double aValue, String aName) { ... }

    public double getMeasure()
    {
        return value;
    }
    ...
}

public class BankAccount implements Measurable
{
    public BankAccount(double initBalance) { ... }

    public double getMeasure()
    {
        return balance;
    }
    ...
}
```

Which of the following statements is correct?

○
```
Measurable m = new Coin(0.1, "dime");
Coin dime = m;
```

Correct!

●
```
Coin dime = new Coin(0.1, "dime");
Measurable m = dime;
```

○
```
Coin dime = new Coin(0.1, "dime");
BankAccount b = (Measureable)dime;
```

○
```
Measureable m = new BankAccount(1000);
BankAccount b = m;
```

Question 8

1 / 1 pts

If you have multiple classes in your program that have implemented the same interface in different ways, how is the correct method executed?

○ The method must be qualified with the class name to determine the correct method.

Correct!

● The Java virtual machine must locate the correct method by looking at the class of the actual object.

○ You cannot have multiple classes in the same program with different implementations of the same interface.

○ The compiler must determine which method implementation to use.

Question 9

1 / 1 pts

Consider the following class:

```
public class Player implements Comparable
{
    private String name;
    private int goalsScored;

    // other methods go here

    public int compareTo(Object otherObject)
    {
        __________________________________
        return goalsScored - otherPlayer.goalsScored;
    }
}
```

What statement can be used to complete the `compareTo()` method?

Correct!

● `Player otherPlayer = (Player)otherObject;`

○ `Player otherPlayer = otherObject;`

○ `Object otherPlayer = (Player)otherObject;`

○ `Object otherPlayer = otherObject;`

Question 10

1 / 1 pts

A method that has no implementation is called a/an _______ method.

○ interface

Correct!

● abstract

○ overloaded

○ implementation

Quiz Score: **10** out of 10