Quiz - Results

✕

Attempt 1 of 1

Written Oct 11, 2023 10:42 AM - Oct 11, 2023 11:45 AM

Attempt Score	**40 / 65 - 61.54 %**
Overall Grade (Highest Attempt)	**40 / 65 - 61.54 %**

Question 4 — 1 / 2 points

Which of the following statements are true?

- ✗ □ A subclass is a subset of a superclass.
- ✓ □ "class A extends B" means B is a subclass of A.
- ⇨ ✓ □ "class A extends B" means A is a subclass of B.
- ⇨ ✓ □ A subclass is usually extended to contain more functions and more detailed information than its superclass.

▼ Hide question 4 feedback

Well done!

Well done!

Question 5 — 0 / 1 point

Why is a static variable also referred to as a class variable?

- ⇨ ○ There is a single copy available to all objects of the class.
- ✗ ○ It is encapsulated within the class.
- ○ Each class has one and only one static variable.

○ It is stored in the separate class area of each object.

Question 6 0 / 1 point

Can the method `lastDayOfMonth` be changed to be a static method by just changing the header to the following?

```
private static int lastDayOfMonth()
```

➡ ○ No. It could no longer access the `month` instance variable.

○ No. A class cannot have both static and non-static methods.

✗ ○ Yes. No other change is necessary.

○ Yes, but it must be changed to public as well.

Question 11 0 / 1.5 points

Consider the following code snippet:

```
public class Motorcycle extends Vehicle
{
   . . .
   public Motorcycle(int numberAxles)
   {
      super(numberAxles); //line #1
   }
}
```

If the line marked "//`line #1`" was missing, which of these statements would be correct?

○ This code would not compile.

○ The `Motorcycle` class constructor would invoke the constructor of the `Vehicle` class with a parameter value of 0.

➡ ○ The `Motorcycle` class constructor would invoke the constructor of the `Vehicle` class with no parameters.

✗ ○ The `Vehicle` class constructor would invoke the constructor of the `Motorcycle` class with no parameters.

Question 12 0 / 2.5 points

Consider the following code snippet:

```
public class Vehicle
```

```
{
   . . .
   public void setVehicleClass(double numberAxles)
   {
      . . .
   }
}
public class Auto extends Vehicle
{
   . . .
   public void setVehicleClass(int numberAxles)
   {
      . . .
   }
}
```

Which of the following statements is correct?

○ The `Vehicle` class overloads the `setVehicleClass` method.

✗ ○ The `Auto` class overrides the `setVehicleClass` method.

○ The `Vehicle` class overrides the `setVehicleClass` method.

⇨ ○ The `Auto` class overloads the `setVehicleClass` method.

▼ Hide question 12 feedback

Incorrect

Question 13 0 / 1 point

Which of interface methods that are not abstract?

○ A) no static methods

○ B) public methods

⇨ ○ C) static and default methods

✗ ○ D) private and protected methods

Question 14 0 / 2.5 points

Consider the following class hierarchy:

```
public class Vehicle
{
   private String type;
   public Vehicle(String type)
   {
      this.type = type;
   }
   public String displayInfo()
   {
      return type;
   }
}

public class LandVehicle extends Vehicle
{
   public LandVehicle(String type)
   {
      super(type);
   }
}

public class Auto extends LandVehicle
{
   public Auto(String type)
   {
      super(type);
   }
}
```

You have written a program to use these classes, as shown in the following code snippet:

```
public class VehicleTester
{
   public static void main(String[] args)
   {
      Auto myAuto = new Auto("sedan");
      System.out.println("MyAuto type = " + ______);
   }
}
```

Complete the code in this program snippet to correctly display the auto's type.

○ This cannot be done unless the `Auto` class overrides the `displayInfo` method.

➡ ○ `myAuto.displayInfo()`

✗ ○ `myAuto.super.super.displayInfo()`

○ `myAuto.super.displayInfo()`

▼ Hide question 14 feedback

Incorrect

Question 16 0 / 1 point

The use of the `static` keyword in a method declaration implies which of the following?

○ The method can only operate on immutable objects.

○ The method can only be called from within the main method.

✗ ○ The method cannot be overloaded.

⇨ ○ The method cannot be invoked on an instance of an object.

Question 19 0 / 1 point

Which of the following statements about abstract methods is true?

⇨ ○ An abstract method has a name, parameters, and a return type, but no code in the body of the method.

✗ ○ An abstract method has only a name and a return type, but no parameters or code in its body.

○ An abstract method has parameters, a return type, and code in its body, but has no defined name.

○ An abstract method has a name, a return type, and code in its body, but has no parameters.

▼ Hide question 19 feedback

Incorrect

Question 21 0 / 2 points

Consider the classes shown below:

```
public class Parent
{
   public int getValue()
   {
      return 24;
   }
   public void display()
   {
      System.out.print(getValue() + " ");
   }
}
public class Child extends Parent
{
   public int getValue()
   {
      return -7;
   }
}
```

Using the classes above, what is the output of the following lines of code?

```
Parent kid = new Child();
Parent adult = new Parent();
kid.display();
adult.display();
```

✗ ○ 24 24

○ -7 -7

○ 24 -7

➡ ○ -7 24

Question 26

0 / 1 point

Assume a class implements two interfaces, both of which define a default method with the same signature. Which statement is true about this conflict of inherited methods?

○ The code compiles and the implementation is chosen at run time.

○ The code compiles but generates an exception at run time due to the conflict.

➡ ○ The class must override the method and provide its own implementation.

✗ ○ There is no conflict because interfaces cannot provide method implementation.

▼ Hide question 26 feedback

Incorrect

Question 30

0 / 1 point

A theater needs a `TicketCounter` to keep track of the number of tickets sold. There are two types of ticket: regular and discount. What instance data should be used for this class?

○ `private String[] ticketsSold; //  Each entry is either "regular" or "discount"`

➡ ○ `private int regularTicketsSold;`
`int discountTicketsSold;`

✗ ○ `private ArrayList<String> ticketsSold; //  Each entry is either "regular" or "discount"`

○ `private double ticketsSold; //  Add 1 for regular and 0.5 for discount tickets`

Question 31

0 / 1 point

Identify the association between MyCalendar and Day:

```
public class MyCalendar
{
    enum Day
    {
      FRIDAY,SATURDAY,SUNDAY,MONDAY,
      TUESDAY,WEDNESDAY,THURSDAY
    }
}
```

○ A) IS-A

✗ ○ B) HAS-A

➡ ○ C) OWNS-A

○ D) No association

Question 32

0 / 1.5 points

Suppose the abstract class `Message` is defined below

```
public abstract class Message
{
   private String value;
   public Message(String initial)
   {
      value = initial;
   }

   public String getMessage()
   {
      return value;
   }

   public abstract String translate();
}
```

A concrete subclass of `Message`, called `FrenchMessage`, is defined. Which methods must `FrenchMessage` define?

✗ ○ The `FrenchMessage` constructor and `translate()` only

○ The `FrenchMessage` constructor, `getMessage()`, and `translate()`

➡ ○ `translate()` only

○ `getMessage()` only

Question 36 0 / 1 point

Consider the partial class below:

```
public class Thing
{
   private int number;
   private char letter;
   {
      number = -10;
      letter = 'Z';
   }
 ...
}
```

Which code is equivalent to the code above?

➡ ○

```
public class Thing
{
   private int number = -10;
   private char letter = 'Z';
   ...
}
```

○

```
public class Thing
{
   private int number;
   private char letter;
   public Thing()
   {
      number = -10;
      letter = 'Z';
   }
   ...
}
```

○
```
public class Thing
{
   public Thing()
   {
      int number;
      char letter;
      number = -10;
      letter = 'Z';
   }
   ...
}
```

✖ ○
```
public class Thing
{
   public static void main(String[] args)
   {
      int number;
      char letter;
      number = -10;
      letter = 'Z';
   }
   ...
}
```

Question 39 0 / 2 points

Suppose the abstract class `Message` is defined below

```
public abstract class Message
{
   private String value;
   public Message(String initial)
   {
      value = initial;
   }

   public String getMessage()
   {
      return value;
   }

   public abstract String translate();
}
```

A concrete subclass of `Message`, called `FrenchMessage`, is defined. Which methods must `FrenchMessage` define?

○ `getMessage()` only

➡ ○ `translate()` only

✗ ○ The `FrenchMessage` constructor and `translate()` only

○ The `FrenchMessage` constructor, `getMessage()`, and `translate()`

▼ Hide question 39 feedback

Incorrect

Question 44 **0 / 1 point**

Which of the following Object-Oriented (OO) concepts is best described as the capability of a subclass to provide different implementation for a method that is already defined and/or implemented in its superclass or one of its parent superclasses?

✗ ○ Polymorphism

○ Method overloading

○ Abstraction

➡ ○ Method overriding

Question 47 **0 / 1 point**

Identify primitive types in the following Java program:

```
public class TestingMemory
{
   public static void main(String[] args){

      double x=Math.PI;
      String y="Burnaby Mountain";
      Object z=new Point(49.2667,122.9667);
      int w=123;
   }
}
```

○ A) x and z

➡ ○ B) x and w

✖ ○ C) y and z

○ D) x and y

Question 50 **0 / 1 point**

By default all enum constructors are ________.

○ A) public or private

➡ ○ B) private

✖ ○ C) public

○ D) protected

Done