

University of the People  
Practice Quizzes Units 4-6  
Course: CS 1102

Consider the following Java program, which one of the following best describes "pepper"?

Consider the following Java program, which one of the following best describes "pepper"?

```
public class Food {  
    static int count;  
    private String flavor = "sweet";  
    Food() { count++; }  
    void setFlavor(String s) { flavor = s; }  
    String getFlavor() { return flavor; }  
    static public void main(String[] args) {  
        Food pepper = new Food();  
        System.out.println(pepper.getFlavor());  
    }  
}
```

Select one:

- ☐ a. a method
- ☐ b. a class variable
- ☒ c. a local object variable
- ☐ d. an instance variable
- ☐ e. a constructor



Your answer is correct.

See Section 5.1.2 of Eck (2014).

The correct answer is: a local object variable

Consider the following Java program, which one of the following best describes "pepper"?

Consider the following Java program, which one of the following best describes "pepper"?

```
public class Food {  
    static int count;  
    private String flavor = "sweet";  
    Food() { count++; }  
    void setFlavor(String s) { flavor = s; }  
    String getFlavor() { return flavor; }  
    static public void main(String[] args) {  
        Food pepper = new Food();  
        System.out.println(pepper.getFlavor());  
    }  
}
```

Select one:

- ☐ a. an object
- ☒ b. a reference ✓
- ☐ c. null
- ☐ d. sweet
- ☐ e. The program does not compile.

What is the output of the following Java program?

What is the output of the following Java program?

```
class Food {  
    String flavor = "bland";  
}  
class Pepper extends Food {  
    String flavor = "spicy";  
}  
public class Lunch {  
    public static void main(String[] args) {  
        Food lunch = new Pepper();  
        System.out.println(lunch.flavor);  
    }  
}
```

Select one:

- ☒ a. bland
- ☐ b. no output
- ☐ c. spicy
- ☐ d. the program does not compile
- ☐ e. bland



spicy

Your answer is correct.

A variable of type "Food" sees the member "flavor" from the "Food" class, even though the object also has a member "flavor" from the "Pepper" class. Methods are polymorphic, but member variables are not. See Sections 5.5.3 and 5.6.2.

The correct answer is: bland

Considering the following Java sample program, what are the "x" and "y" values?

Considering the following Java sample program, what are the "x" and "y" values?

```
public void mouseDragged(MouseEvent evt) {  
    if ( dragging == false )  
        return;  
    int x = evt.getX();  
    int y = evt.getY();  
    .  
    . . .  
    prevX = x;  
    prevY = y;  
}
```

- 
- ☐ a. Next position of Mouse
  - ☐ b. Position of Mouse when clicking
  - ☐ c. Position of Mouse after dragging
  - ☐ d. Position of Mouse during the drag
  - ☒ e. Current position of Mouse



Your answer is correct.

See Section 6.3.3

The correct answer is:  
Current position of Mouse

Which of the following is used to interact with GUI components?

Which of the following is used to interact with GUI components?

---

- ☐ a. Keyboard
- ☐ b. Mouse and keyboard
- ☒ c. All above are correct
- ☐ d. Other input devices
- ☐ e. Mouse



Your answer is correct.

The correct answer is:  
All above are correct

What is the output of the following Java program?

What is the output of the following Java program?

```
class Food {  
    void flavor() { System.out.println("bland"); }  
}  
class Pepper extends Food {  
    void flavor() { System.out.println("spicy"); }  
}  
public class Lunch {  
    public static void main(String[] args) {  
        Pepper lunch = new Food();  
        lunch.flavor();  
    }  
}
```

Select one:

- ☐ a. bland  
spicy
- ☒ b. the program does not compile
- ☐ c. no output
- ☐ d. bland
- ☐ e. spicy



Your answer is correct.

The program tries to initialize "lunch", a variable of type "Pepper", with an object of type "Food". A superclass variable can be initialized to a subclass object, but not the other way around. See Section 5.5.3.

The correct answer is: the program does not compile

What is the output of the following Java program?

What is the output of the following Java program?

```
class Food { String flavor = "bland"; }
class Pepper extends Food {
    String flavor = "spicy";
    Pepper(String flavor) { this.flavor = flavor; }
}
public class Lunch {
    public static void main(String[] args) {
        Pepper pepper = new Pepper("sweet");
        System.out.println(pepper.flavor);
    }
}
```

Select one:

- ☐ a. spicy
- ☐ b. no output
- ☒ c. sweet
- ☐ d. the program does not compile
- ☐ e. bland



Your answer is correct.

The "Pepper" constructor uses "this" to set the "flavor" member variable to the "flavor" parameter. See Section 5.6.1.

What is the output of the following Java program?

What is the output of the following Java program?

```
public class Food {  
    static int count;  
    private String flavor = "sweet";  
    Food(String s) { flavor = s; }  
    void setFlavor(String s) { flavor = s; }  
    String getFlavor() { return flavor; }  
    static public void main(String[] args) {  
        Food pepper = new Food("spicy");  
        Food chile = new Food("spicy");  
        System.out.println(pepper == chile);  
    }  
}
```

Select one:

- ☐ a. smoky
- ☐ b. sweet
- ☐ c. true
- ☐ d. spicy
- ☒ e. false



Comparing object variables with "==" compares the values of the memory locations of the objects, not the values inside the objects. Because "pepper" and "chile" get the results from different "new" calls, they store different memory locations. See Section 5.1.2 of Eck (2014).

Consider the following Java program, which one of the following best describes "flavor"?



Consider the following Java program, which one of the following best describes "flavor"?

```
public class Food {  
    static int count;  
    private String flavor = "sweet";  
    Food() { count++; }  
    void setFlavor(String s) { flavor = s; }  
    String getFlavor() { return flavor; }  
    static public void main(String[] args) {  
        Food pepper = new Food();  
        System.out.println(pepper.getFlavor());  
    }  
}
```

---

Select one:

- ☐ a. a constructor
- ☐ b. a local object variable
- ☒ c. an instance variable
- ☐ d. a class variable
- ☐ e. a method



Your answer is correct.

See Section 5.1.1 of Eck (2014).

The correct answer is: an instance variable

What is the output of the following Java program?

```
interface Food {  
    public void printFlavor();  
}  
class Pepper implements Food {  
    public void printFlavor() {  
System.out.println("spicy"); }  
}  
public class Lunch {  
    public static void main(String[] args) {  
        Food pepper = new Pepper();  
        pepper.printFlavor();  
    }  
}
```

Select one:

- ☐ a. the program does not compile
- ☒ b. spicy
- ☐ c. bland
- ☐ d. no output
- ☐ e. bland



spicy

Your answer is correct.

The "Pepper" class implements the "Food" interface. In particular, it defines the "printFlavor" method, which prints "spicy". See Section 5.7.1.

The correct answer is: spicy

What is the output of the following Java program?

```
class Food {  
    Food() { System.out.println("bland"); }  
    Food(String flavor) { System.out.println(flavor); }  
}  
class Pepper extends Food {  
    Pepper() { super("spicy"); }  
}  
public class Lunch {  
    public static void main(String[] args) {  
        Food lunch = new Pepper();  
    }  
}
```

Select one:

- ☐ a. the program does not compile
- ☐ b. bland  
spicy
- ☒ c. spicy
- ☐ d. bland
- ☐ e. no output



Your answer is correct.

Because the "Pepper" constructor explicitly calls 'super("spicy")', the "Food" constructor that takes a String parameter gets called instead of the "Food" constructor that takes no parameters. See Section 5.6.3.


The correct answer is: spicy

Considering the following Java code, what is the "a" parameter?

```
Color myColor = Color.color(r, g, b, a);
```

Considering the following Java code, what is the "a" parameter?

```
Color myColor = Color.color(r, g, b, a);
```

- 
- ☐ a. It defines the Brightness value
  - ☐ b. It's a wrong parameter and the "Color" can be defined with only (r, g, b)
  - ☐ c. It defines the Gama value
  - ☒ d. It defines the Opaque value 
  - ☐ e. It defines the fourth color value

Your answer is correct.

See Section 6.2.1

The correct answer is:


It defines the Opaque value

Which one of the following claims about Java is INCORRECT?

Which one of the following claims about Java is INCORRECT?

---

Select one:

- ☐ a. A constructor's name must match the name of the class.
- ☒ b. A default constructor is always provided by the compiler. 
- ☐ c. Default and no arguments constructor is not the same.
- ☐ d. Constructors can be overloaded with different parameter lists.
- ☐ e. A constructor's name is its return type.

Your answer is correct.

The compiler only provides a default constructor if a class does not have any explicit constructor definitions. See Section 5.2.2 of Eck (2014).

The correct answer is: A default constructor is always provided by the compiler.

Consider the following Java program, which one of the following best describes "setFlavor"?

Consider the following Java program, which one of the following best describes "setFlavor"?

```
public class Food {  
    static int count;  
    private String flavor = "sweet";  
    Food() { count++; }  
    void setFlavor(String s) { flavor = s; }  
    String getFlavor() { return flavor; }  
    static public void main(String[] args) {  
        Food pepper = new Food();  
        System.out.println(pepper.getFlavor());  
    }  
}
```

Select one:

- ☐ a. gets executed when the program runs
- ☒ b. a setter method
- ☐ c. a static method
- ☐ d. a class method
- ☐ e. a public method



Your answer is correct.

See Section 5.1.3 of Eck (2014).

The correct answer is: a setter method

What is the output of the following Java program?

What is the output of the following Java program?

```
public class Food {
    static int count;
    private String flavor = "sweet";
    Food(String s) { flavor = s; }
    void setFlavor(String s) { flavor = s; }
    String getFlavor() { return flavor; }
    static public void main(String[] args) {
        Food pepper = new Food("spicy");
        Food chile = pepper;
        pepper.setFlavor("smoky");
        System.out.println(chile.getFlavor());
    }
}
```

Select one:

- ☒ a. smoky
- ☐ b. spicy
- ☐ c. true
- ☐ d. sweet
- ☐ e. false



Your answer is correct.


Setting "chile = pepper" causes the two object variables to refer to the same single object. Because "chile" and "pepper" refer to the same object, the flavor set by "pepper.setFlavor" is seen by "chile.getFlavor". See Section 5.1.2 of Eck (2014).

The correct answer is: smoky

What is the JavaFX application thread?

What is the JavaFX application thread?

---

- ☐ a. JavaFX application thread is one of the GUI library components
- ☒ b. When the main() routine is executed, the launch() method creates a new thread, called the JavaFX application thread 
- ☐ c. When the main() routine is executed, the JavaFX application thread can invoke
- ☐ d. JavaFX application thread is used for running Java applications
- ☐ e. JavaFX application thread is a virtual thread to control GUI applications

Your answer is correct.

See Section 6.1.1

The correct answer is:

When the main() routine is executed, the launch() method creates a new thread, called the JavaFX application thread

What is the output of the following Java program?



What is the output of the following Java program?

```
class Food {
    String flavor = "bland";
    void printFlavor() { System.out.println(flavor); }
}
class Pepper extends Food {
    String flavor = "spicy";
}
public class Lunch {
    public static void main(String[] args) {
        Food lunch = new Pepper();
        lunch.printFlavor();
    }
}
```

Select one:

- ☐ a. bland  
spicy
- ☒ b. bland
- ☐ c. no output
- ☐ d. spicy



Your answer is correct.

The "Pepper" class inherits the "printFlavor" method but does not override it. Because member variables are NOT polymorphic, the "printFlavor" method defined in "Food" sees the member variable "flavor" in "Food", not the one in "Pepper". See Sections 5.6.1 and 5.6.2.

The correct answer is: bland

What is the output of the following Java program?

What is the output of the following Java program?

```
public class Food {  
    static int count;  
    private String flavor = "sweet";  
    Food() { count++; }  
    void setFlavor(String s) { flavor = s; }  
    String getFlavor() { return flavor; }  
    static public void main(String[] args) {  
        Food pepper = new Food();  
        new Food();  
        System.out.println(pepper.count);  
    }  
}
```

Select one:

- ☒ a. 2
- ☐ b. The program does not compile.
- ☐ c. sweet
- ☐ d. spicy
- ☐ e. 1



The first call to "new Food()" increments the class variable "count". The second call to "new Food()" increments it again, for a value of 2. Notice that you do not have to use the resulting reference from a "new" call. Also notice that objects (like "pepper") have access to the class variables of their class (like "count"). See Sections 5.1.2 and 5.2.2 of Eck (2014).

A subclass method can \_\_\_\_ a superclass method with the same name and parameter types.

A subclass method can \_\_\_\_\_ a superclass method with the same name and parameter types.

---

Select one:

- ☒ a. override
- ☐ b. abstract
- ☐ c. inherit
- ☐ d. implement
- ☐ e. extend



Your answer is correct.

See Section 5.6.2.

The correct answer is: override

What is the output of the following Java Program?

What is the output of the following Java program?

```
class Food {  
    Food() { System.out.println(flavor); }  
    String flavor = "bland";  
}  
class Pepper extends Food {  
    String flavor = "spicy";  
}  
public class Lunch {  
    public static void main(String[] args) {  
        Food lunch = new Pepper();  
    }  
}
```

Select one:

- ☐ a. no output
- ☒ b. bland
- ☐ c. spicy
- ☐ d. bland  
spicy

- ☐ e. the program does not compile

Unlike methods, constructors are NOT polymorphic. The compiler automatically calls the "Food" constructor, which uses the "flavor" member for class "Food", not the one for the subclass "Pepper". See Sections 5.6.2 and 5.6.3.

An object that registers listener objects does which of the following?

An object that registers listener objects does which of the following?

---

Select one:

- ☐ a. It maintains an object directory.
- ☐ b. It records audio.
- ☐ c. It handles events.
- ☒ d. It generates events.
- ☐ e. It runs an event loop.



Your answer is correct.

See Section 6.3.1.

The correct answer is: It generates events.