Oracle

1Z0-808
Java SE 8 Programmer I

For More Information – Visit link below:

https://www.examsboost.com/

Product Version

✓ **Up to Date products, reliable and verified.**
✓ **Questions and Answers in PDF Format.**

Latest Version: 20.0

Question: 1

Given:

```
class Product {
    double price;
}

public class Test {
    public void updatePrice(Product product, double price) {
        price = price * 2;
        product.price = product.price + price;
    }
    public static void main(String[] args) {
        Product prt = new Product();
        prt.price = 200;
        double newPrice = 100;

        Test t = new Test();
        t.updatePrice(prt, newPrice);
        System.out.println(prt.price + " : " + newPrice);
    }
}
```

What is the result?

A. 200.0 : 100.0
B. 400.0 : 200.0
C. 400.0 : 100.0
D. Compilation fails.

Answer: C

Question: 2

Which statement is true about the switch statement?

A. It must contain the default section.
B. The break statement, at the end of each case block, is optional.
C. Its case label literals can be changed at runtime.

D. Its expression must evaluate to a collection of values.

Answer: B

Question: 3

Given the code fragment:

```
public static void main(String[] args) {
    String date = LocalDate
                    .parse("2014-05-04")
                    .format(DateTimeFormatter.ISO_DATE_TIME);
    System.out.println(date);
}
```

What is the result?

A. May 04, 2014T00:00:00.000
B. 2014-05-04T00:00: 00.000
C. 5/4/14T00:00:00.000
D. An exception is thrown at runtime.

Answer: D

Question: 4

Given the code fragment:

```
public static void main(String[] args) {
    Short s1 = 200;
    Integer s2 = 400;
    Long s3 = (long) s1 + s2;        //line n1
    String s4 = (String) (s3 * s2);     //line n2
    System.out.println("Sum is " + s4);
}
```

What is the result?

A. Sum is 600
B. Compilation fails at line n1.
C. Compilation fails at line n2.
D. A ClassCastException is thrown at line n1.
E. A ClassCastException is thrown at line n2.

Question: 5

What is the name of the Java concept that uses access modifiers to protect variables and hide them within a class?

A. Encapsulation
B. Inheritance
C. Abstraction
D. Instantiation
E. Polymorphism

Answer: A

Using the private modifier is the main way that an object encapsulates itself and hide data from the outside world.

Question: 6

Given the code fragment:

```
abstract class Planet {
    protected void revolve() {              //line n1
    }

    abstract void rotate();                 //line n2
}

class Earth extends Planet {
    void revolve() {                        //line n3
    }

    protected void rotate() {               //line n4
    }
}
```

Which two modifications, made independently, enable the code to compile? (Choose two.)

A. Make the method at line n1 public.

B. Make the method at line n2 public.
C. Make the method at line n3 public.
D. Make the method at line n3 protected.
E. Make the method at line n4 public.

Answer: C,D

Question: 7

Given:

```
class Vehicle {
    String type = "4W";
    int maxSpeed = 100;

    Vehicle(String type, int maxSpeed) {
        this.type = type;
        this.maxSpeed = maxSpeed;
    }
    Vehicle() {}
}

class Car extends Vehicle {
    String trans;

    Car(String trans) {                //line n1
        this.trans = trans;
    }

    Car(String type, int maxSpeed, String trans) {
        super(type, maxSpeed);    // line n2
        this.trans = trans;
    }
}
```

And given the code fragment:

```
 7. Car c1 = new Car("Auto");
 8. Car c2 = new Car("4W", 150, "Manual");
 9. System.out.println(c1.type + " " + c1.maxSpeed + " " + c1.trans);
10. System.out.println(c2.type + " " + c2.maxSpeed + " " + c2.trans);
```

What is the result?

A. 4W 100 Auto4W 150 Manual
B. null 0 Auto4W 150 Manual
C. Compilation fails only at line n1
D. Compilation fails only at line n2
E. Compilation fails at both line n1 and line n2

Answer: E

Question: 8

Given:

```
class Caller {
    private void init () {
        System.out.println("Initialized");
    }

    private void start () {
    init();
    System.out.println("Started");
    }
}

public class TestCall {
    public static void main(String[] args) {
        Caller c = new Caller();
        c.start(); // line n1
        c.init();  // line n2
    }
}
```

What is the result?

A. Compilation fails at line n1.
B. InitializedStartedInitialized
C. InitializedStarted
D. Compilation fails at line n2.

Answer: D

Question: 9

Given these two classes:

```
public class Customer {
    ElectricAccount acct = new ElectricAccount();

    public void useElectricity(double kWh){
        acct.addKWh(kWh);
    }
}

public class ElectricAccount {
    private double kWh;
    private double rate = 0.07;
    private double bill;

    //line n1
}
```

Any amount of electricity used by a customer (represented by an instance of the Customer class) must contribute to the customer's bill (represented by the member variable bill) through the useElectricity method.
An instance of the Customer class should never be able to tamper with or decrease the value of the member variable bill.
How should you write methods in the ElectricAccount class at line n1 so that the member variable bill is always equal to the value of the member variable kwh multiplied by the member variable rate?

A

```
    public void addKWh(double kWh) {
        this.kWh += kWh;
        this.bill = this.kWh*this.rate;
    }
```

B

```
    public void addKWh(double kWh) {
        if (kWh > 0){
            this.kWh += kWh;
            this.bill = this.kWh * this.rate;
        }
    }
```

C

```
    private void addKWh(double kWh) {
        if (kWh > 0) {
            this.kWh += kWh;
            this.bill = this.kWh*this.rate;
        }
    }
```

D

```
    public void addKWh(double kWh) {
        if(kWh > 0) {
            this.kWh += kWh;
            setBill(this.kWh);
        }
    }
    public void setBill(double kWh) {
        bill = kWh*rate;
    }
```

A. Option A
B. Option B
C. Option C
D. Option D

Answer: A

Question: 10

Given the code fragment:

```
public static void main(String[] args) {
    StringBuilder sb = new StringBuilder("Java");
    String s = "Java";

    if (sb.toString().equals(s.toString()))  {
        System.out.println("Match 1");
    } else if (sb.equals(s)) {
        System.out.println("Match 2");
    } else {
        System.out.println("No Match");
    }
}
```

What is the result?

A. Match 1
B. Match 2
C. No Match
D. A NullPointerException is thrown at runtime.

Answer: A

Thank You for Trying Our Product