

## Free PassLeader 1Z0-808 Exam Dumps with VCE and PDF Download (Question 49 - Question 56)

PassLeader now are offering 100% pass ensure 1Z0-808 dumps! All 1Z0-808 exam questions have been updated with correct answers, welcome to download the newest PassLeader 1Z0-808 VCE dumps and PDF dumps:

<http://www.passleader.com/1z0-808.html> (256 Q&As --> 312 Q&As)

BTW: Download PassLeader 1Z0-808 dumps from Google Drive for free:

[https://drive.google.com/open?id=0B-ob6L\\_QjGLpc2VwdWJ4RFVZYkE](https://drive.google.com/open?id=0B-ob6L_QjGLpc2VwdWJ4RFVZYkE)

### QUESTION 49

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);  
}
```

[www.passleader.com](http://www.passleader.com)

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

Explanation:

The exception `java.time.temporal.UnsupportedTemporalTypeException` is thrown at runtime.

We should use class `LocalDateTime` with `ISO_DATE_TIME` format or use the format `ISO_DATE` to avoid the exception.

See `ISO_DATE_TIME` at

<https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html>

See examples at

<https://gist.github.com/mscharhag/9195718>

### QUESTION 50

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);  
}
```

[www.passleader.com](http://www.passleader.com)

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.

Answer: C

Explanation:

Compilation fails at n2 because the compiler cannot cast long to String.

#### QUESTION 51

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

Explanation:

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

[http://www.tutorialspoint.com/java/java\\_access\\_modifiers.htm](http://www.tutorialspoint.com/java/java_access_modifiers.htm)

#### QUESTION 52

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?

- 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: CD

Explanation:

We can't assign weaker privileges in a subclass.

Method revolve() is declared protected in class Planet.

We can declare revolve() as public or protected in class Earth.

#### QUESTION 53

Given:

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

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

class Car extends Vehicle {
    String trans;

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

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

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);
www.passleader.com
```

What is the result?

- A. 4W 100 Auto  
4W 150 Manual
- B. Null 0 Auto  
4W 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

Explanation:

Compilation fails at n1 because Vehicle hasn't a default constructor

Compilation fails at n2 because this() must be the first statement in constructor body

QUESTION 54

Given the code fragment:

```
1. class X { www.passleader.com
2.     public void printFileContent() {
3.         /* code goes here */
4.         throw new IOException();
5.     }
6. }
7. public class Test {
8.     public static void main(String[] args) {
9.         X xobj = new X();
10.        xobj.printFileContent();
11.    }
12. }
```

Which two modifications should you make so that the code compiles successfully?

- A) Replace line 8 with `public static void main(String[] args) throws Exception {`
- B) Replace line 10 with:

```
try {  
    xobj.printFileContent();  
}  
catch(Exception e) { }  
catch(IOException e) { }
```
- C) Replace line 2 with `public void printFileContent() throws IOException {`
- D) Replace line 4 with `throw IOException("Exception raised");`
- E) At line 11, insert `throw new IOException();` [www.passleader.com](http://www.passleader.com)

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

Answer: AC

Explanation:

Add throws clause in both `printFileContent` and `main`.

#### QUESTION 55

Given the following 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  
}
```

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`? 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 method `useElectricity` method. An instance of the customer class should never be able to tamper with or decrease the value of the member variable `bill`.

- 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;
}
```
- [www.passleader.com](http://www.passleader.com)

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

Answer: D

#### QUESTION 56

Given the code fragments:

Person.java:

[www.passleader.com](http://www.passleader.com)

```
public class Person {
    String name;
    int age;

    public Person(String n, int a) {
        name = n;
        age = a;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }
}
```

Test.java:

```
public static void checkAge(List<Person> list, Predicate<Person> predicate) {
    for (Person p : list) {
        if (predicate.test(p)) {
            System.out.println(p.name + " ");
        }
    }
}

public static void main(String[] args) {
    List<Person> iList = Arrays.asList(new Person("Hank", 45),
                                      new Person("Charlie", 40),
                                      new Person("Smith", 38));
    //line n1
}
```

Which code fragment, when inserted at line n1, enables the code to print Hank?

- A. `checkAge (iList, () -> p. get Age ( ) > 40);`
- B. `checkAge(iList, Person p -> p.getAge( ) > 40);`
- C. `checkAge (iList, p -> p.getAge ( ) > 40);`
- D. `checkAge(iList, (Person p) -> { p.getAge() > 40; });`

Answer: C

Explanation:

<https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html>

PassLeader now are offering 100% pass ensure 1Z0-808 dumps! All 1Z0-808 exam questions have been updated with correct answers, welcome to download the newest PassLeader 1Z0-808 VCE dumps and PDF dumps:

<http://www.passleader.com/1z0-808.html> (256 Q&As --> 312 Q&As)

BTW: Download PassLeader 1Z0-808 dumps from Google Drive for free:

[https://drive.google.com/open?id=0B-ob6L\\_QjGLpc2VwdWJ4RFVZYkE](https://drive.google.com/open?id=0B-ob6L_QjGLpc2VwdWJ4RFVZYkE)