Mphasis-PreAsessment-Java

0% (0/15)

✗ **1.**

```
public abstract class Shape {
private int x;
private int y;
public abstract void draw();
public void setAnchor(int x, int y) {
this.x = x;
this.y = y;
}
}
```

Which two classes use the Shape class correctly? (Choose two.)

Ⓐ

```
public class Circle implements Shape {
private int radius;
}
```

Ⓑ

```
public abstract class Circle extends Shape {
private int radius;
}
```

Ⓒ

```
public class Circle extends Shape {
private int radius;
public void draw();
}
```

Ⓓ

```
public abstract class Circle implements Shape {
private int radius;
public void draw();
}
```

Ⓔ

```
public class Circle extends Shape {
private int radius;
public void draw() {/* code here */}
}
```

Ⓕ

```
public abstract class Circle implements Shape {
private int radius;
public void draw() {/* code here */}
}
```

✗ 2. **Which statement is true about the classes and interfaces in the exhibit?**

```
01. public interface A {
02. public void doSomething(String thing);
03. }
01. public class AImpl implements A {
02. public void doSomething(String msg) {}
03. }
01. public class B {
02. public A doit(){
03. //more code here
04. }
05. public String execute(){
06 //more code here
07 }
08. }
01. public class C extends B {
02. public AImpl doit(){
03. //more code here
04. }
05.
06. public Object execute() {
07. //more code here
08. }

09. }
```

Ⓐ Compilation will succeed for all classes and interfaces.

Ⓑ Compilation of class C will fail because of an error in line 2.

Ⓒ Compilation of class C will fail because of an error in line 6.

Ⓓ Compilation of class AImpl will fail because of an error in line 2.

✗ 3. **Given:**

```
public static void parse(String str) {
try {
float f = Float.parseFloat(str);
} catch (NumberFormatException nfe) {
f = 0;
} finally {
System.out.println(f);
}
}
public static void main(String[] args) {
parse("invalid");
}
```

What is the result?

Ⓐ 0.0

Ⓑ Compilation fails.

Ⓒ A ParseException is thrown by the parse method at runtime.

Ⓓ A NumberFormatException is thrown by the parse method at runtime.

✗ **4. Given:**

```
01. public class Blip {
02. protected int blipvert(int x) { return 0; }
03. }
04. class Vert extends Blip {
05. // insert code here
06. }
```

Which five methods, inserted independently at line 5, will compile? (Choose five.)

Ⓐ public int blipvert(int x) { return 0; }

Ⓑ private int blipvert(int x) { return 0; }

Ⓒ private int blipvert(long x) { return 0; }

Ⓓ protected long blipvert(int x) { return 0; }

Ⓔ protected int blipvert(long x) { return 0; }

Ⓕ protected long blipvert(long x) { return 0; }

Ⓖ protected long blipvert(int x, int y) { return 0; }

✗ **5.**

```
public class TestString1 {
public static void main(String[] args) {
String str = "420";
str += 42;
System.out.print(str);
}
}
```

What is the output?

Ⓐ 42

Ⓑ 420

Ⓒ 462

Ⓓ 42042

Ⓔ Compilation fails.

Ⓕ An exception is thrown at runtime.

✗ 6. Given:

```
23. Object [] myObjects = {
24. new Integer(12),
25. new String("foo"),
26. new Integer(5),
27. new Boolean(true)
28. };
29. Arrays.sort(myObjects);
30. for(int i=0; i<myObjects.length; i++) {
31. System.out.print(myObjects[i].toString());
32. System.out.print(" ");
33. }
```

What is the result?

- (A) Compilation fails due to an error in line 23.
- (B) Compilation fails due to an error in line 29.
- (C) A ClassCastException occurs in line 29.
- (D) A ClassCastException occurs in line 31.
- (E) The value of all four objects prints in natural order.

✗ 7. Which statement is true?

- (A) A class's finalize() method CANNOT be invoked explicitly.
- (B) super.finalize() is called implicitly by any overriding finalize() method.
- (C) The finalize() method for a given object is called no more than once by the garbage collector.
- (D) The order in which finalize() is called on two objects is based on the order in which the two objects became
  finalizable.

✗ 8. package com;

```
public class Test {

private int empId;
private String empName;
public String designation;
public int getEmpId() {
return empId;
}
public void setEmpId(int empId) {
this.empId = empId;
}
public String getEmpName() {
return empName;
}
public void setEmpName(String empName) {
this.empName = empName;
}
public String getDesignation() {
return designation;
}
public void setDesignation(String designation) {
```

```
this.designation = designation;
}
}
```

Above defined class is not fully encapsulated. why?

○ A all methods defined as public

○ B all properties defined as private

○ C designation property is defined as public

○ D Test class is defined as public

✗ 9.

```
class Employee{
@Override
public void finalize(){
System.out.println("Finallize method got called");
}
}
class Test{
@Override
public void finalize(){
System.out.println("Finallize method got called");
}
public static void main(String[] args){

Employee emp=new Employee();
String str=new String("Abc");
System.gc();
}
}
```

Select One correct option

○ A Finalize method of Employee executed

● B Finalize method of Test executed

○ C None of classes Finalize method gets called

○ D Finalize method cannot be overridden in Test class. Because Test is not sub class of Employee

✗ 10. interface DoStuff2 {

```
float getRange(int low, int high);
}
interface DoMore {
float getAvg(int a, int b, int c);
}
abstract class DoAbstract implements DoStuff2, DoMore {
}
06. class DoStuff implements DoStuff2 {
07. public float getRange(int x, int y) {
08. return 3.14f;
09. }
10. }
11.
12. interface DoAll extends DoMore {
13. float getAvg(int a, int b, int c, int d);
14. }
```

What is the result?

Ⓐ The file will compile without error.

Ⓑ Compilation fails. Only line 7 contains an error.

Ⓒ Compilation fails. Only line 12 contains an error.

Ⓓ Compilation fails. Only line 13 contains an error.

Ⓔ Compilation fails. Only lines 7 and 12 contain errors.

✗ 11. What is displayed on the console when running the following program?

```
class Test {

public static void main(String[] args) {

try {

method();

System.out.println("After the method call");

}

catch (NumberFormatException ex) {

System.out.println("NumberFormatException");

}

catch (RuntimeException ex) {

System.out.println("RuntimeException");
}

}

static void method() {

String s = "5.6";

Integer.parseInt(s); // Cause a NumberFormatException

int i = 0;

int y = 2 / i;

System.out.println("Welcome to Java");

}

}
```

Ⓐ The program displays NumberFormatException.

Ⓑ The program displays NumberFormatException followed by After the method call.

Ⓒ The program has a compilation error.

Ⓓ The program displays RuntimeException.

✗ 12.

```
public class BuildStuff {
public static void main(String[] args) {
Boolean test = new Boolean(true);
Integer x = 343;
Integer y = new BuildStuff().go(test, x);
System.out.println(y);
}
int go(Boolean b, int i) {
if(b) return (i/7);
return (i/49);
}
}
```

What is the result?

Ⓐ 7

Ⓑ 49

Ⓒ 343

Ⓓ Compilation fails.

Ⓔ An exception is thrown at runtime.

✗ 13. Given:

```
import java.io.*;
public class Forest implements Serializable {
private Tree tree = new Tree();
public static void main(String [] args) {
Forest f = new Forest();
try {
FileOutputStream fs = new FileOutputStream("Forest.ser");
ObjectOutputStream os = new ObjectOutputStream(fs);
os.writeObject(f); os.close();
} catch (Exception ex) { ex.printStackTrace(); }
}
}
class Tree {

}
```

What is the result?

(A) Compilation fails.

(B) An exception is thrown at runtime.

(C) An instance of Forest is serialized.

(D) An instance of Forest and an instance of Tree are both serialized.

✗ 14. C1,C2 and C3 classes defined in various packages as below declaration

```
package p1;
public class C1{

}
package p1.p2;
public class C2{

}
package p1.p2.p3;
class C3{

}
```

what is the correct statement to import class C3?

(A) import p1.*;

(B) import p3.*;

(C) import p1.p2.*;

(D) import p1.p3.p2.*;

(E) import p1.p2.p3.*;

✗ 15. Analyze the following code:

```
class Test {
public static void main(String[] args) {
try {
String s = "5.6";
Integer.parseInt(s); // Cause a NumberFormatException

int i = 0;
int y = 2 / i;
}
catch (Exception ex) {
System.out.println("NumberFormatException");
}
catch (RuntimeException ex) {
System.out.println("RuntimeException");
}
}
}
```

Ⓐ The program displays NumberFormatException.

Ⓑ The program displays RuntimeException.

Ⓒ The program displays NumberFormatException followed by RuntimeException.

Ⓓ The program has a compilation error.