

CTS-JAVA-SE Test

Total Questions: 17

Most Correct Answers: #17

Least Correct Answers: #16

```
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.)

- 0/25 ☐ A public class Circle implements Shape {
 private int radius;
}
- 25/25 ☒ B public abstract class Circle extends Shape {
 private int radius;
}
- 0/25 ☐ C public class Circle extends Shape {
 private int radius;
 public void draw();
}
- 0/25 ☐ D public abstract class Circle implements Shape {
 private int radius;
 public void draw();
}
- 25/25 ☒ E public class Circle extends Shape {
 private int radius;
 public void draw() { /* code here */ }
}
- 0/25 ☐ F 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 Almpl 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 Almpl doit(){  
03.         //more code here  
04.     }  
05.     public Object execute() {  
06.         //more code here  
07.     }  
08. }
```

09. }

- 0/25 ☐ A Compilation will succeed for all classes and interfaces.
- 4/25 ☐ B Compilation of class C will fail because of an error in line 2.
- 21/25 ☒ C Compilation of class C will fail because of an error in line 6.
- 0/25 ☐ D Compilation of class Almpl 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?

- 0/25 ☐ A 0.0
- 24/25 ☒ B Compilation fails.
- 1/25 ☐ C A ParseException is thrown by the parse method at runtime.
- 0/25 ☐ D 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.)

- 25/25 ☒ A public int blipvert(int x) { return 0; }
- 1/25 ☐ B private int blipvert(int x) { return 0; }
- 25/25 ☒ C private int blipvert(long x) { return 0; }
- 3/25 ☐ D protected long blipvert(int x) { return 0; }
- 25/25 ☒ E protected int blipvert(long x) { return 0; }
- 25/25 ☒ F protected long blipvert(long x) { return 0; }
- 21/25 ☒ G 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?

- 0/25 ☐ A 42
- 0/25 ☐ B 420
- 0/25 ☐ C 462
- 25/25 ☒ D 42042
- 0/25 ☐ E Compilation fails.
- 0/25 ☐ F 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?

- 0/25 ☐ A Compilation fails due to an error in line 23.
- 0/25 ☐ B Compilation fails due to an error in line 29.
- 24/25 ☒ C A ClassCastException occurs in line 29.
- 0/25 ☐ D A ClassCastException occurs in line 31.
- 1/25 ☐ E The value of all four objects prints in natural order.

7. Which statement is true?

- 0/25 ☐ A A class's finalize() method CANNOT be invoked explicitly.
- 0/25 ☐ B super.finalize() is called implicitly by any overriding finalize() method.
- 25/25 ☒ C The finalize() method for a given object is called no more than once by the garbage collector.
- 0/25 ☐ D The order in which finalize() is called on two objects is based on the order in which the two objects became finalizable.

```

8.  import java.util.*;
    public class Mapit {
    public static void main(String[] args) {
    Set<Integer> set = new HashSet<Integer>();
    Integer i1 = 45;
    Integer i2 = 46;
    set.add(i1);
    set.add(i1);
    set.add(i2); System.out.print(set.size() + " ");
    set.remove(i1); System.out.print(set.size() + " ");
    i2 = 47;
    set.remove(i2); System.out.print(set.size() + " ");
    }
    }

```

What is the result?

- 0/25 ☐ A 2 1 0
- 25/25 ☒ B 2 1 1
- 0/25 ☐ C 3 2 1
- 0/25 ☐ D 3 2 2
- 0/25 ☐ E Compilation fails.
- 0/25 ☐ F An exception is thrown at runtime.

```

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

- 0/25 ☐ A Finalize method of Employee executed
- 0/25 ☐ B Finalize method of Test executed
- 22/25 ☒ C No classes Finalize method got called
- 3/25 ☐ 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?

```

- 24/25 ☒ A The file will compile without error.
- 0/25 ☐ B Compilation fails. Only line 7 contains an error.
- 0/25 ☐ C Compilation fails. Only line 12 contains an error.
- 0/25 ☐ D Compilation fails. Only line 13 contains an error.
- 1/25 ☐ E Compilation fails. Only lines 7 and 12 contain errors.

11. What is the output of this program?

```

class Test {
int a;
public int b;
private int c;
}
class AccesTest {
public static void main(String args[])
{
Test ob = new Test();
ob.a = 10;
ob.b = 20;
ob.c = 30;
System.out.println(" Output :a, b, and c" + ob.a + " " + ob.b + " " + ob.c);
}
}

```

- 25/25 ☒ A Compilation error
- 0/25 ☐ B Run time error
- 0/25 ☐ C Output : a, b and c 10 20 30
- 0/25 ☐ D None of the mentioned

```

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?

- 0/25 ☐ A 7
- 21/25 ☒ B 49
- 0/25 ☐ C 343
- 4/25 ☐ D Compilation fails.
- 0/25 ☐ E An exception is thrown at runtime.

13. Given:

```

import java.io.*;
public class Forest implements Serializable {

```

```

    public static void main(String [] args) {
        Tree t = new Tree();
        try {
            FileOutputStream fs = new FileOutputStream("Forest.ser");
            ObjectOutputStream os = new ObjectOutputStream(fs);
            os.writeObject(t);
            os.close();
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
class Tree {

```

What is the result?

- 3/25 ☐ A Compilation fails.
- 22/25 ☒ B An exception is thrown at runtime.
- 0/25 ☐ C An instance of Forest is serialized.
- 0/25 ☐ D An instance of Forest and an instance of Tree are both serialized.

```
14. class Test{
public static void main(String[] args){
String test = "This is a test";
12. String[] tokens = test.split("\s");
13. System.out.println(tokens.length);
}
}
```

What is the result?

- 0/25 ☐ A 0
- 0/25 ☐ B 1
- 0/25 ☐ C 4
- 25/25 ☒ D Compilation fails.

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");
}
}
}
```

- 1/25 ☐ A The program displays NumberFormatException.
- 2/25 ☐ B The program displays RuntimeException.
- 7/25 ☐ C The program displays NumberFormatException followed by RuntimeException.
- 15/25 ☒ D The program has a compilation error.


```
16. Given:
20. public class CreditCard {
21.
22. private String cardID;
23. private Integer limit;
24. public String ownerName;
25.
26. public void setCardInformation(String cardID,
27. String ownerName,
28. Integer limit) {
29. this.cardID = cardID;
30. this.ownerName = ownerName;
31. this.limit = limit;
32. }
33. }
```

Select one correct option from following

- 0/25 ☐ A The class is fully encapsulated.
- 0/25 ☐ B The code demonstrates polymorphism.
- 25/25 ☐ C The ownerName variable breaks encapsulation.
- 0/25 ☒ D The cardID and limit variables break polymorphism.
- 0/25 ☐ E The setCardInformation method breaks encapsulation.

```
17. Given:
11. public interface Status {
12. /* insert code here */ int MY_VALUE = 10;
13. }
```

Which three are valid on line 12? (Choose three.)

- 25/25 ☒ A final
- 25/25 ☒ B static
- 0/25 ☐ C native
- 25/25 ☒ D public
- 0/25 ☐ E private
- 0/25 ☐ F abstract
- 0/25 ☐ G protected