socrative by MasteryConnect

CTS-JAVA-SE Test

Total Questions: 17

Most Correct Answers: **#11**

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.)

0/16 (A)
```
public class Circle implements Shape {
private int radius;
}
```

8/16 (B)
```
public abstract class Circle extends Shape {
private int radius;
}
```

2/16 (C)
```
public class Circle extends Shape {
private int radius;
public void draw();
}
```

5/16 (D)
```
public abstract class Circle implements Shape {
private int radius;
public void draw();
}
```

15/16 (E)
```
public class Circle extends Shape {
private int radius;
public void draw() {/* code here */}
}
```

0/16 (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 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. }
```

6/16 (A) Compilation will succeed for all classes and interfaces.

4/16 (B) Compilation of class C will fail because of an error in line 2.

2/16 (C) Compilation of class C will fail because of an error in line 6.

2/16 (D) 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?

5/16 (A) 0.0

5/16 (B) Compilation fails.

3/16 (C) A ParseException is thrown by the parse method at runtime.

1/16 (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.)

9/16 (A) public int blipvert(int x) { return 0; }

6/16 (B) private int blipvert(int x) { return 0; }

9/16 (C) private int blipvert(long x) { return 0; }

9/16 (D) protected long blipvert(int x) { return 0; }

13/16 (E) protected int blipvert(long x) { return 0; }

14/16 (F) protected long blipvert(long x) { return 0; }

9/16 (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?

0/16 (A) 42

0/16 (B) 420

1/16 (C) 462

9/16 (D) 42042

4/16 (E) Compilation fails.

0/16 (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?

0/16 (A) Compilation fails due to an error in line 23.

8/16 (B) Compilation fails due to an error in line 29.

3/16 (C) A ClassCastException occurs in line 29.

1/16 (D) A ClassCastException occurs in line 31.

2/16 (E) The value of all four objects prints in natural order.

7. Which statement is true?

2/16 (A) A class's finalize() method CANNOT be invoked explicitly.

3/16 (B) super.finalize() is called implicitly by any overriding finalize() method.

8/16 (C) The finalize() method for a given object is called no more than once by the garbage collector.

1/16 (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?

6/16 (A) 2 1 0

4/16 (B) 2 1 1

1/16 (C) 3 2 1

0/16 (D) 3 2 2

1/16 (E) Compilation fails.

2/16 (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

6/16 (A) Finalize method of Employee executed

2/16 (B) Finalize method of Test executed

5/16 (C) No classes Finalize method got called

1/16 (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?

8/16 (A) The file will compile without error.

2/16 (B) Compilation fails. Only line 7 contains an error.

0/16 (C) Compilation fails. Only line 12 contains an error.

3/16 (D) Compilation fails. Only line 13 contains an error.

1/16 (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 AcessTest {
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);
}
}
```

14/16 (A) Compilation error

0/16 (B) Run time error

0/16 (C) Output : a, b and c 10 20 30

0/16 (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?

2/16 A 7

4/16 B 49

0/16 C 343

8/16 D Compilation fails.

0/16 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?

1/16 A Compilation fails.

5/16 B An exception is thrown at runtime.

7/16 C An instance of Forest is serialized.

1/16 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?

0/16 A 0

0/16 B 1

6/16 C 4

8/16 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");
}
}
}
```

6/16 A The program displays NumberFormatException.

3/16 B The program displays RuntimeException.

1/16 C The program displays NumberFormatException followed by RuntimeException.

4/16 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

0/16 A The class is fully encapsulated.

0/16 B The code demonstrates polymorphism.

13/16 C The ownerName variable breaks encapsulation.

1/16 D The cardID and limit variables break polymorphism.

0/16 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.)

11/16 A final

10/16 B static

1/16 C native

13/16 D public

3/16 E private

0/16 F abstract

4/16 G protected