|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 1 / 1 point |

Under the rules of inheritance, private data fields in the superclass are

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | not accessible in the subclass | | | |
|  | |  |  | | --- | --- | |  | protected in the subclass | | | |
|  | |  |  | | --- | --- | |  | public in the subclass | | | |
|  | |  |  | | --- | --- | |  | accessible through polymorphism only | | | |
| **Question 2** | |  | 1 / 1 point | |

A subclass (or derived class)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | incorporates data fields and methods from its superclass and can have additional fields and methods as well | | | |
|  | |  |  | | --- | --- | |  | incorporates data fields and methods from its superclass but cannot have additional fields and methods | | | |
|  | |  |  | | --- | --- | |  | incorporates data fields , but not methods, from its superclass and can have additional fields or methods as well | | | |
|  | |  |  | | --- | --- | |  | incorporates data fields, but not methods, from its superclass but cannot have additional fields or methods | | | |
| **Question 3** | |  | 1 / 1 point | |

Given these class definitions:

public class A {

public A(int x) {

System.out.print(x);

}

}

public class B extends A {

public B() {

super(1);

System.out.print("B");

}

}

what is printed out by the following Java statement:

A a = new B();

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | 1B | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
|  | |  |  | | --- | --- | |  | xB | | | |
|  | |  |  | | --- | --- | |  | 1 | | | |
| **Question 4** | |  | 1 / 1 point | |

Inheritance refers to the ability to

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | redefine how member methods of related classes operate | | | |
|  | |  |  | | --- | --- | |  | create new classes from existing ones | | | |
|  | |  |  | | --- | --- | |  | combine data fields and methods into one type | | | |
|  | |  |  | | --- | --- | |  | none of the above | | | |
| **Question 5** | |  | 1 / 1 point | |

The following prototype shows that a Cylinder subclass is derived from a superclass called Circle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | class Circle extends Cylinder | | | |
|  | |  |  | | --- | --- | |  | class Cylinder derived Circle | | | |
|  | |  |  | | --- | --- | |  | class Cylinder extends Circle | | | |
|  | |  |  | | --- | --- | |  | class Circle derived Cylinder | | | |
| **Question 6** | |  | 1 / 1 point | |

Under the rules of inheritance, public methods in the superclass are

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | public in the subclass | | | |
|  | |  |  | | --- | --- | |  | protected in the subclass | | | |
|  | |  |  | | --- | --- | |  | accessible in the subclass | | | |
|  | |  |  | | --- | --- | |  | accessible through polymorphism only | | | |
| **Question 7** | |  | 1 / 1 point | |

In the following Java code fragment:

public class MyClass {

public void a() {

int p;

p = 98;

}

public void b() {

a();

/\* state of p here?\*/

}

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | p will have value 98 | | | |
|  | |  |  | | --- | --- | |  | p will have value 0 | | | |
|  | |  |  | | --- | --- | |  | p will not exist | | | |
|  | |  |  | | --- | --- | |  | p will have garbage value | | | |
| **Question 8** | |  | 1 / 1 point | |

What is true of the following Java code?

int[] array = new int[10];

for (int i = array.length; i <= 0; i--){

array[i] = 50;

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | It would generate a runtime error | | | |
|  | |  |  | | --- | --- | |  | it would result in an array with ten elements of value 0 | | | |
|  | |  |  | | --- | --- | |  | it would assign all of the elements of array the value 50 | | | |
|  | |  |  | | --- | --- | |  | it would result in an infinite loop | | | |
| **Question 9** | |  | 1 / 1 point | |

Given the following Java code:

public class A {

private int x = 30;

}

public class B extends A {

public void b(){

System.out.print(x);

}

}

what would be the result of the statements:

B b = new B();

b.b();

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | 30 | | | |
|  | |  |  | | --- | --- | |  | nothing, because the code would not compile | | | |
|  | |  |  | | --- | --- | |  | 0 | | | |
|  | |  |  | | --- | --- | |  | x | | | |
| **Question 10** | |  | 1 / 1 point | |

Which statement is true: With inheritance, a derived subclass object can directly access any

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | public or private superclass members |
|  | | | private superclass members |
|  | | | public or protected superclass members |
|  | | | protected, public or private superclass member |
|  |  |
|  | |

|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 1 / 1 point |

Given these class definitions:

public class A {

public A() {

System.out.print("A");

}

}

public class B extends A {

public B() {

System.out.print("B");

}

}

Which of the following statements would result in a runtime error?

|  |  |  |  |
| --- | --- | --- | --- |
|  | A a = (A) new B(); | | |
|  | A a = new A(); | | |
|  | A a = new B(); | | |
|  | A a = (B) new A(); | | |
| **Question 2** | |  | 1 / 1 point | |

What is true of the following Java code?

int[] array = new int[10];

for (int i = 0; i < array.length; i--){

array[i] = 50;

}

|  |  |  |  |
| --- | --- | --- | --- |
|  | it would assign all of the elements of array the value 50 | | |
|  | it would result in a compile-time error | | |
|  | it would result in an IndexOutOfBoundsException being thrown | | |
|  | it would generate a runtime error | | |
|  | it would assign all of the elements of array the value 0 | | |
| **Question 3** | |  | 1 / 1 point | |

What is true of the following Java code?

int[] array = new int[10];

for (int i = array.length-1; i >= 0; i--){

array[i] = 50;

}

|  |  |  |  |
| --- | --- | --- | --- |
|  | it would assign all of the elements of array the value 50 | | |
|  | it would assign all of the elements of array the value 0 | | |
|  | it would generate a runtime error | | |
|  | it would result in an IndexOutOfBoundsException being thrown | | |
|  | it would assign all of the elements of array the value 50 except for the last that will have 0 | | |
| **Question 4** | |  | 1 / 1 point | |

Given these class definitions:

public class A {

public void doIt() {

System.out.print("a");

}

}

public class B extends A {

public void doIt() {

System.out.print("b");

}

}

What is printed out by the following Java statements:

A a = new B();

a.doIt();

|  |  |  |  |
| --- | --- | --- | --- |
|  | a | | |
|  | b | | |
|  | ab | | |
|  | ba | | |
|  | None of the above | | |
| **Question 5** | |  | 1 / 1 point | |

Given these class definitions:

public class A {

public A() {

System.out.print("A");

}

}

public class B extends A {

public B() {

this(4);

System.out.print("B");

}

public B(int x){

System.out.print(x);

}

}

What is printed out by the following Java statement:

B b = new B();

|  |  |  |  |
| --- | --- | --- | --- |
|  | AB | | |
|  | 4AB | | |
|  | A4B | | |
|  | AB4 | | |
|  | None of the above | | |
| **Question 6** | |  | 1 / 1 point | |

What is result of the following Java code?

int[][] array = new int[5][2];

System.out.print(array[0].length);

|  |  |  |  |
| --- | --- | --- | --- |
|  | 2 | | |
|  | 5 | | |
|  | 10 | | |
|  | 0 | | |
|  | None of the above | | |
| **Question 7** | |  | 1 / 1 point | |

Given these class definitions:

public class A {

public void a() {

System.out.print("A");

}

}

public class B extends A {

public B() {

System.out.print("B");

}

}

What is printed out by the following Java statement:

A a = new B();

|  |  |  |  |
| --- | --- | --- | --- |
|  | A | | |
|  | B | | |
|  | AB | | |
|  | BA | | |
|  | none of the above | | |
| **Question 8** | |  | 1 / 1 point | |

Given these class definitions:

public class A {

public A(int x) {

System.out.print("A"+x);

}

}

public class B extends A {

public B() {

System.out.print("B");

}

}

What is printed out by the following Java statement:

A a = new B();

|  |  |  |  |
| --- | --- | --- | --- |
|  | A0 | | |
|  | B | | |
|  | A0B | | |
|  | BA0 | | |
|  | none of the above | | |
| **Question 9** | |  | 1 / 1 point | |

What is result of the following Java code?

int[][] array = {  
 { 1, 2, 3},  
 { 4, 5, 6},  
 { 7, 8, 9}  
 };  
 int[] row = array[0];

System.out.print(row[1]);

|  |  |  |  |
| --- | --- | --- | --- |
|  | 2 | | |
|  | compiler error | | |
|  | 5 | | |
|  | 4 | | |
|  | ArrayIndexOutOfBoundsException | | |
| **Question 10** | |  | 1 / 1 point | |

Given these class definitions:

public class A {

public A() {

System.out.print("A");

}

}

public class B extends A {

public B() {

System.out.print("B");

}

}

What is printed out by the following Java statement:

A a = new B();

|  |  |  |  |
| --- | --- | --- | --- |
|  | A | | |
|  | B | | |
|  | AB | | |
|  | BA | | |
|  | None of the above | | |
| **Question 11** | |  | 1 / 1 point | |

What is result of the following Java code?

int[][] array = new int[5][2];

System.out.print(array.length);

|  |  |  |  |
| --- | --- | --- | --- |
|  | 2 | | |
|  | 5 | | |
|  | 10 | | |
|  | 0 | | |
|  | None of the above | | |
| **Question 12** | |  | 1 / 1 point | |

What is true of the following Java code?

int[] array = new int[10];

for (int i = 0; i > array.length; i--){

array[i] = 50;

}

|  |  |  |  |
| --- | --- | --- | --- |
|  | it would result in an IndexOutOfBoundsException being thrown | | |
|  | it would generate a runtime error | | |
|  | it would result in a compile-time error | | |
|  | it would assign all of the elements of array the value 50 | | |
|  | it would assign all of the elements of array the value 0 | | |
| **Question 13** | |  | 1 / 1 point | |

Given these class definitions:

public class A {

public A(int x) {

System.out.print(x);

}

}

public class B extends A {

public B() {

super(1);

System.out.print("B");

}

}

What is printed out by the following Java statement:

A a = new B();

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1B | | |
|  | xB | | |
|  | 1 | | |
|  | B | | |
|  | None of the above | | |
| **Question 14** | |  | 1 / 1 point | |

What is result of the following Java code?

int[][] array = {  
 { 1, 2, 3},  
 { 4, 5, 6},  
 { 7, 8, 9}  
 };

System.out.print(array[1][2]);

|  |  |  |  |
| --- | --- | --- | --- |
|  | 2 | | |
|  | 8 | | |
|  | ArrayIndexOutOfBoundsException | | |
|  | 4 | | |
|  | 6 | | |
| **Question 15** | |  | 1 / 1 point | |

What is true of the following Java code?

int[] array = new int[10];

for (int i = array.length; i <= 0; i--){

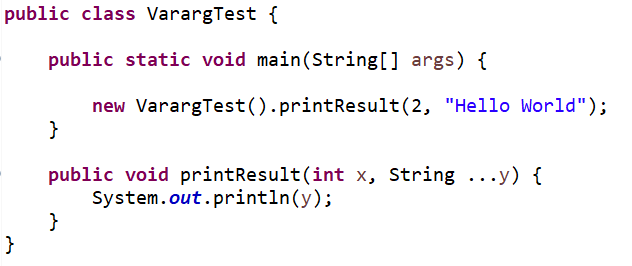
array[i] = 50;

}

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | it would generate a runtime error |
|  | | | it would assign all of the elements of array the value 0 |
|  | | | it would assign all of the elements of array the value 50 |
|  | | | it would result in an IndexOutOfBoundsException being thrown |
|  | | | it would result in a compile-time error |
|  |  |
|  | |

|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 1 / 1 point |

What will be the output of the following code segment:

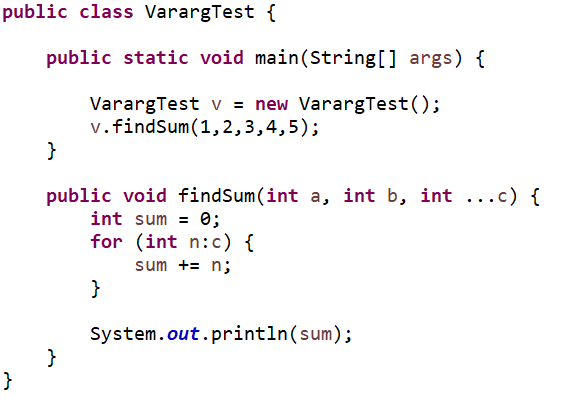


|  |  |  |  |
| --- | --- | --- | --- |
|  | Hello World | | |
|  | None of these, but printing the reference | | |
|  | Hello | | |
|  | H | | |
| **Question 2** | |  | 1 / 1 point | |

Varargs can be used *only* in the final argument position

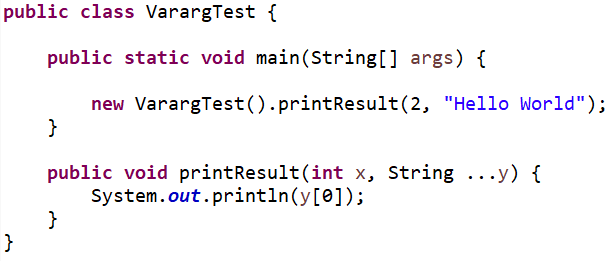
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | True | | |
|  |  | False | | |
| **Question 3** | | |  | 1 / 1 point | |

What will the following code print:



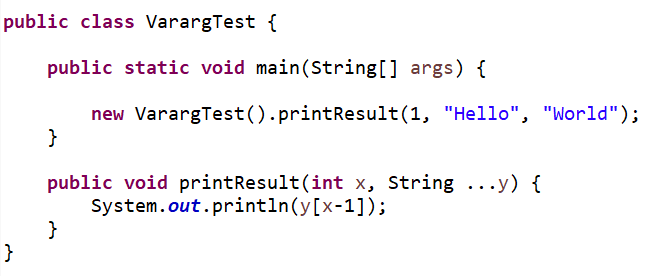
|  |  |  |  |
| --- | --- | --- | --- |
|  | 15 | | |
|  | 5 | | |
|  | 12 | | |
|  | 9 | | |
| **Question 4** | |  | 1 / 1 point | |

What is the output of the following code:



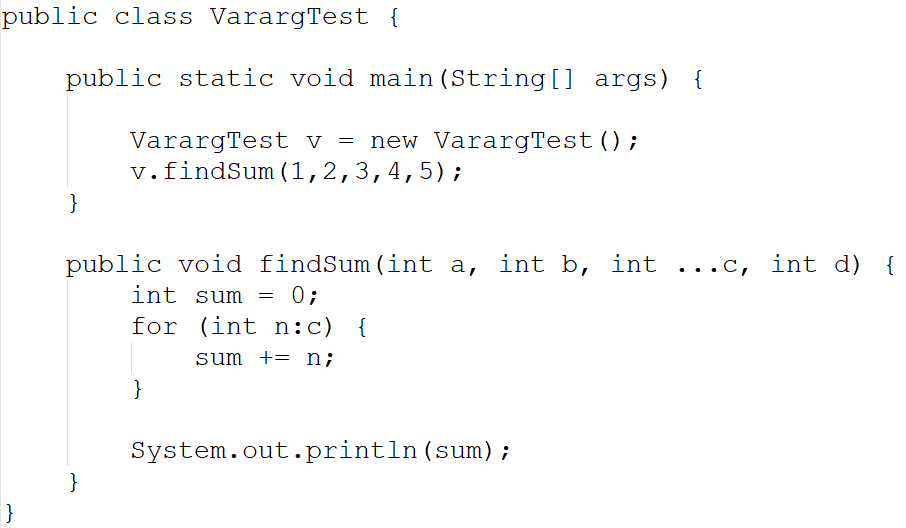
|  |  |  |  |
| --- | --- | --- | --- |
|  | none of these, but printing reference | | |
|  | Hello World | | |
|  | H | | |
|  | Hello | | |
| **Question 5** | |  | 1 / 1 point | |

What will be the output of the following code segment:



|  |  |  |  |
| --- | --- | --- | --- |
|  | Hello | | |
|  | H | | |
|  | Hello World | | |
|  | None of these, but printing the reference | | |
| **Question 6** | |  | 1 / 1 point | |

What will be the output of the following code:



|  |  |
| --- | --- |
|  | 12 |
|  | 15 |
|  | 7 |
|  | None of these |

|  |  |  |
| --- | --- | --- |
| **Question 1**Correct on previous attempt(s) |  | 1 / 1 point |

Which of the following statements about Java Abstract Classes and Interfaces are true?

|  |  |  |  |
| --- | --- | --- | --- |
|  | all of these answers | | |
|  | all attributes in an interface, if any, must be constants (i.e. public static final) | | |
|  | a class can extend only one abstract class, but a class can implement more than one interface | | |
|  | an interface doesn't include implementation (method bodies) for any of its method definitions | | |
| **Question 2**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions:

**public class A {**

**public A() {**

**System.out.print("A");**

**}**

**}**

**public class B extends A {**

**public B() {**

**System.out.print("B");**

**}**

**}**

which of the following statements would result in an error?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | **A a = new A();** | | | |
|  | |  |  | | --- | --- | |  | **A a = new B();** | | | |
|  | |  |  | | --- | --- | |  | **A a = (A) new B();** | | | |
|  | |  |  | | --- | --- | |  | **A a = (B) new A();** | | | |
| **Question 3**Correct on previous attempt(s) | |  | 1 / 1 point | |

What would be printed by executing the following Java code?

**ArrayList<int> myList;**

**myList = new ArrayList<int>();**

**System.out.print(myList.size());**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | Nothing, the code will not compile. | | | |
|  | |  |  | | --- | --- | |  | **10** | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | **-1** | | | |
|  | |  |  | | --- | --- | |  | **0** | | | |
| **Question 4**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions:

**public class A {**

**public A(int x) {**

**System.out.print(x);**

**}**

**}**

**public class B extends A {**

**public B() {**

**super(1);**

**System.out.print("B");**

**}**

**}**

what is printed out by the following Java statement:

**A a = new B();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **1B** | | | |
|  | |  |  | | --- | --- | |  | **xB** | | | |
|  | |  |  | | --- | --- | |  | **1** | | | |
|  | |  |  | | --- | --- | |  | **B** | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
| **Question 5**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given the following Java code

**public class A {**

**protected int x = 3;**

**}**

**public class B extends A {**

**public void printIt(){**

**System.out.print(x);**

**}**

**}**

what would be the result of the statements

**B b = new B();**

**b.printIt();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **3** would be printed | | | |
|  | |  |  | | --- | --- | |  | None of the these answers | | | |
|  | |  |  | | --- | --- | |  | nothing, because the code would not compile | | | |
|  | |  |  | | --- | --- | |  | **X** would be printed | | | |
| **Question 6**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions:

**public class A {**

**public A() {**

**System.out.print("A");**

**}**

**}**

**public class B extends A {**

**public void doit() {**

**System.out.print("doit");**

**}**

**}**

what is printed out by the following Java statement:

**A b = new B();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **A** | | | |
|  | |  |  | | --- | --- | |  | **doitA** | | | |
|  | |  |  | | --- | --- | |  | **doit** | | | |
|  | |  |  | | --- | --- | |  | **Adoit** | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
| **Question 7**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given the following class definition:

**public class A {**

**int x;**

**int y;**

**public A() {**

**x = 1;**

**y = 7;**

**}**

**public void doIt(){**

**int x = 5;**

**System.out.print(x);**

**System.out.print(y);**

**}**

**}**

what would be printed by the following Java code?

**A a = new A();**

**a.doIt();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **15** | | | |
|  | |  |  | | --- | --- | |  | **17** | | | |
|  | |  |  | | --- | --- | |  | **51** | | | |
|  | |  |  | | --- | --- | |  | **57** | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
| **Question 8**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given the following class definition:

**public class A {**

**int x = 4;**

**public A() {**

**int y = 0;**

**}**

**public void aMethod(){**

**System.out.print(x);**

**}**

**}**

what would be printed by the following Java code?

**A a = new A();**

**a.aMethod();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
|  | |  |  | | --- | --- | |  | **44** | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | **0** | | | |
|  | |  |  | | --- | --- | |  | **4** | | | |
| **Question 9**Correct on previous attempt(s) | |  | 1 / 1 point | |

What is true of the following Java code?

**int[] array = new int[10];**

**for (int i = array.length - 1 ; i >= 0; i--){**

**array[i] = 50;**

**}**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | it would generate a runtime error | | | |
|  | |  |  | | --- | --- | |  | it would result in an infinite loop | | | |
|  | |  |  | | --- | --- | |  | it would assign all of the elements of **array** the value **50** | | | |
|  | |  |  | | --- | --- | |  | it would result in a compile time error | | | |
|  | |  |  | | --- | --- | |  | it would print **50** ten times | | | |
| **Question 10**Correct on previous attempt(s) | |  | 1 / 1 point | |

What is the output of the following Java code fragment?

ArrayList<Integer> list1 = new ArrayList<Integer>();

list1.add(new Integer(1));

list1.add(new Integer(2));

list1.add(new Integer(3));

list1.remove(1);

System.out.println(list1);

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **[1, 2, 3]** | | | |
|  | |  |  | | --- | --- | |  | **[1]** | | | |
|  | |  |  | | --- | --- | |  | **[1, 3]** | | | |
|  | |  |  | | --- | --- | |  | **None of the above** | | | |
| **Question 11**Correct on previous attempt(s) | |  | 1 / 1 point | |

What is the output of the following Java code fragment:

**ArrayList <String>obj = new ArrayList<String>();**

**obj.add("A");**

**obj.ensureCapacity(3);**

**System.out.println(obj.size());**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **1** | | | |
|  | |  |  | | --- | --- | |  | **2** | | | |
|  | |  |  | | --- | --- | |  | **3** | | | |
|  | |  |  | | --- | --- | |  | **4** | | | |
| **Question 12**Correct on previous attempt(s) | |  | 1 / 1 point | |

What is true of class variables in Java that are declared to be static?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | they are immutable and cannot be changed | | | |
|  | |  |  | | --- | --- | |  | they are a property of every object that is an instance of the class | | | |
|  | |  |  | | --- | --- | |  | they are associated with the class in which they are declared as opposed to objects that are instances of the class | | | |
|  | |  |  | | --- | --- | |  | they are implemented on the stack | | | |
| **Question 13**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions:

**public class A {**

**public void doit() {**

**System.out.print("a");**

**}**

**}**

**public class B extends A {**

**public void doit() {**

**System.out.print("b");**

**}**

**}**

what is printed out by the following Java statements:

**A a = new B();**

**a.doit();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **ba** | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
|  | |  |  | | --- | --- | |  | **ab** | | | |
|  | |  |  | | --- | --- | |  | **b** | | | |
|  | |  |  | | --- | --- | |  | **a** | | | |
| **Question 14**Correct on previous attempt(s) | |  | 1 / 1 point | |

Which of the following is true about Objects in Java?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | Local variables implement attributes, and constructors implement behaviors. | | | |
|  | |  |  | | --- | --- | |  | None of these answers. | | | |
|  | |  |  | | --- | --- | |  | Instance variables implement attributes, and methods implement behaviors. | | | |
|  | |  |  | | --- | --- | |  | Instance variables implement attributes, and constructors implement behaviors. | | | |
|  | |  |  | | --- | --- | |  | Local variables implement attributes, and methods implement behaviors. | | | |
| **Question 15**Retaken | |  | 1 / 1 point | |

Immediately after this statement executes, what best describes the result?

**byte[][] x = {1},{2,3,4};**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | None of these answers. | | | |
|  | |  |  | | --- | --- | |  | **x** is a reference variable that now stores the location information for an array of 2 references to 2 arrays of byte values. | | | |
|  | |  |  | | --- | --- | |  | **x** is allocated on the stack; the resulting array is also allocated on the stack. | | | |
|  | |  |  | | --- | --- | |  | **x** is a reference variable that now stores the location information for a two-dimensional array of byte values. | | | |
|  | |  |  | | --- | --- | |  | **x** is a two-dimensional array of byte values. | | | |
| **Question 16**Correct on previous attempt(s) | |  | 1 / 1 point | |

What is the output of the following code fragment:

**ArrayList <String>obj = new ArrayList<String>();**

**obj.add("A");**

**obj.add("B");**

**obj.add("C");**

**obj.add(1, "D");**

**System.out.println(obj);**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **[A, B, C, D]** | | | |
|  | |  |  | | --- | --- | |  | **[A, D, B, C]** | | | |
|  | |  |  | | --- | --- | |  | **[D, A, B, C]** | | | |
|  | |  |  | | --- | --- | |  | **[D, B, C, A]** | | | |
| **Question 17**Correct on previous attempt(s) | |  | 1 / 1 point | |

What would be printed by executing the following Java code?

**ArrayList<Integer> myList;**

**myList = new ArrayList<Integer>();**

**System.out.print(myList.size());**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | **-1** | | | |
|  | |  |  | | --- | --- | |  | Nothing, the code will not compile. | | | |
|  | |  |  | | --- | --- | |  | **10** | | | |
|  | |  |  | | --- | --- | |  | **0** | | | |
| **Question 18**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given the following Java code fragment:

**public static void aMethod(int... args){**

**System.out.print(args[1]);**

**}**

what is the output of the statement

**aMethod(1,2,3);**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **2** | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | **1** | | | |
|  | |  |  | | --- | --- | |  | **3** | | | |
|  | |  |  | | --- | --- | |  | The statement would generate an argument mismatch error | | | |
| **Question 19**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given the following class definition:

**public class A {**

**int y;**

**public A(int x) {**

**int y = x;**

**}**

**public void aMethod(){**

**System.out.print(x);**

**}**

**}**

what would be printed by the following Java code?

**A a = new A(4);**

**a.aMethod();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
|  | |  |  | | --- | --- | |  | **44** | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | **0** | | | |
|  | |  |  | | --- | --- | |  | **4** | | | |
| **Question 20**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions:

**public class A {**

**public void a() {**

**System.out.print("A");**

**}**

**}**

**public class B extends A {**

**public B() {**

**System.out.print("B");**

**}**

**}**

what is printed out by the following Java statement:

**A a = new B();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **B** | | | |
|  | |  |  | | --- | --- | |  | **AB** | | | |
|  | |  |  | | --- | --- | |  | **BA** | | | |
|  | |  |  | | --- | --- | |  | nothing | | | |
|  | |  |  | | --- | --- | |  | **A** | | | |
| **Question 21**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions:

**public class A {**

**public A() {**

**System.out.print("A");**

**}**

**}**

**public class B extends A {**

**public B() {**

**System.out.print("B");**

**}**

**}**

what is printed out by the following Java statement:

**A a = new B();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **AB** | | | |
|  | |  |  | | --- | --- | |  | nothing | | | |
|  | |  |  | | --- | --- | |  | **B** | | | |
|  | |  |  | | --- | --- | |  | **BA** | | | |
|  | |  |  | | --- | --- | |  | **A** | | | |
| **Question 22**Correct on previous attempt(s) | |  | 1 / 1 point | |

 Which of the following is a reason to use an ArrayList instead of an array?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | An ArrayList can grow or shrink as needed, while an array is always the same size. | | | |
|  | |  |  | | --- | --- | |  | You can use a for-each loop on an ArrayList, but not in an array. | | | |
|  | |  |  | | --- | --- | |  | You can store objects in an ArrayList, but not in an array | | | |
|  | |  |  | | --- | --- | |  | **None of the above** | | | |
| **Question 23**Correct on previous attempt(s) | |  | 1 / 1 point | |

Which of the following is true of an abstract class in Java?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | An abstract class can be used as the superclass in polymorphic code. | | | |
|  | |  |  | | --- | --- | |  | An abstract class can only define abstract methods and cannot implement methods. | | | |
|  | |  |  | | --- | --- | |  | An abstract class cannot be extended. | | | |
|  | |  |  | | --- | --- | |  | An abstract class is used to instantiate abstract objects. | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
| **Question 24**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions:

**public class A {**

**public A(int x) {**

**System.out.print("A");**

**}**

**}**

**public class B extends A {**

**public B() {**

**System.out.print("B");**

**}**

**}**

what is printed out by the following Java statement:

**A a = new B();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **AB** | | | |
|  | |  |  | | --- | --- | |  | **BA** | | | |
|  | |  |  | | --- | --- | |  | **A** | | | |
|  | |  |  | | --- | --- | |  | **B** | | | |
|  | |  |  | | --- | --- | |  | this code will not compile | | | |
| **Question 25**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions:

**public class X {**

**int x;**

**public X(){**

**initialize();**

**printHi();**

**}**

**public void printHi(){**

**System.out.print(" X, with x = " + x);**

**}**

**public void initialize(){**

**x = 3;**

**}**

**}**

**public class Y extends X{**

**@Override**

**public void initialize(){**

**x = 99;**

**}**

**@Override**

**public void printHi(){**

**System.out.print(" Y, with x = " + super.x);**

**}**

**}**

what would be printed out by the following two statements?

**new X();**

**new Y();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **X, with x = 99 Y, with x = 99** | | | |
|  | |  |  | | --- | --- | |  | **X, with x = 3 Y, with x = 3** | | | |
|  | |  |  | | --- | --- | |  | **X, with x = 3 X, with x = 3** | | | |
|  | |  |  | | --- | --- | |  | **X, with x = 3 Y, with x = 99** | | | |
|  | |  |  | | --- | --- | |  | **Y, with x = 99 Y, with x = 99** | | | |
| **Question 26**Correct on previous attempt(s) | |  | 1 / 1 point | |

When we are defining a new interface, what happens if we leave out the abstract keyword in the method definitions?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | Because all methods in an interface are abstract, it would cause a compilation error. | | | |
|  | |  |  | | --- | --- | |  | A compilation error would result because abstract methods are not allowed in an interface. | | | |
|  | |  |  | | --- | --- | |  | Because all methods in an interface are abstract, the abstract keyword must not appear. | | | |
| **Question 27**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions:

**public class A {**

**public A() {**

**System.out.print("A");**

**}**

**}**

**public class B extends A {**

**public B() {**

**this(4);**

**System.out.print("B");**

**}**

**public B(int x){**

**System.out.print(x);**

**}**

**}**

what is printed out by the following Java statement:

**B b = new B();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
|  | |  |  | | --- | --- | |  | **A4B** | | | |
|  | |  |  | | --- | --- | |  | **AB4** | | | |
|  | |  |  | | --- | --- | |  | **AB** | | | |
|  | |  |  | | --- | --- | |  | **4AB** | | | |
| **Question 28**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given the following class definition:

**public class A {**

**int x;**

**public A(int y) {**

**int x = y;**

**}**

**public aMethod(){**

**System.out.print(x);**

**}**

**}**

what would be printed by the following Java code?

**A a = new A(4);**

**a.aMethod();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **44** | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | **0** | | | |
|  | |  |  | | --- | --- | |  | **4** | | | |
| **Question 29**Correct on previous attempt(s) | |  | 1 / 1 point | |

What is true of the following Java code?

**int[] array = new int[10];**

**for (int i = 10; i >= 0; i--){**

**array[i] = 50;**

**}**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | it would result in an infinite loop | | | |
|  | |  |  | | --- | --- | |  | it would generate a runtime error | | | |
|  | |  |  | | --- | --- | |  | it would result in a compile time error | | | |
|  | |  |  | | --- | --- | |  | it would assign all of the elements of **array** the value **50** | | | |
|  | |  |  | | --- | --- | |  | it would print **50** ten times | | | |
| **Question 30**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given the following Java code:

**public class A {**

**private int x = 3;**

**}**

**public class B extends A {**

**public void b(){**

**System.out.print(x);**

**}**

**}**

what would be the result of the statements

**B b = new B();**

**b.b();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **3** | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | nothing, because the code would not compile | | | |
|  | |  |  | | --- | --- | |  | **0** | | | |
| **Question 31**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions

**public class A {**

**public A() {**

**System.out.print("A");**

**}**

**}**

**public class B extends A {**

**public B() {**

**System.out.print("B");**

**}**

**}**

what would be printed by the following Java statements?

**A a = new A();**

**if (a instanceof B){**

**System.out.print("B");**

**}else {**

**System.out.print("A");**

**}**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **ABA** | | | |
|  | |  |  | | --- | --- | |  | **ABB** | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
|  | |  |  | | --- | --- | |  | **BB** | | | |
|  | |  |  | | --- | --- | |  | **AA** | | | |
| **Question 32**Correct on previous attempt(s) | |  | 1 / 1 point | |

What is true of the following Java code?

**int[] array = new int[10];**

**for (int i = 0; i < array.length; i=i){**

**array[i] = 50;**

**}**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | it would generate a runtime error | | | |
|  | |  |  | | --- | --- | |  | it would assign all of the elements of **array** the value **50** | | | |
|  | |  |  | | --- | --- | |  | it would result in a compile time error | | | |
|  | |  |  | | --- | --- | |  | it would result in an infinite loop | | | |
|  | |  |  | | --- | --- | |  | it would result in an **IndexOutOfBoundsException** being thrown | | | |
| **Question 33**Correct on previous attempt(s) | |  | 1 / 1 point | |

In the following Java code fragment:

**public class MyClass {**

**int x;**

**public void a() {**

**x = 20;**

**}**

**public void b() {**

**a();**

**/\* state of x here?\*/**

**}**

**}**

what is the best description of the state of the variable **x** at the indicated point?

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **x** will be available to be garbage collected | |
|  | |  |  | | --- | --- | |  | **x** will not exist | |
|  | |  |  | | --- | --- | |  | None of these answers | |
|  | |  |  | | --- | --- | |  | **x** will have value **20** | |
|  | |  |  | | --- | --- | |  | **x** will have no value | |

|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 1 / 1 point |

Given the declaration of an object of class Person called person1; and that Person class has a field called nAge, would we expect the following statement to be allowed: System.out.println( "Age is " + person1.nAge );

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | yes | | |
|  | |  |  | | --- | --- | |  | no | | |
| [View Feedback](javascript://) | |

|  |  |  |
| --- | --- | --- |
| **Question 2** |  | 1 / 1 point |

Given the declaration: Person person1;....A call to displayPerson method would be:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | Person.displayPerson(); | | | |
|  | |  |  | | --- | --- | |  | person1.displayPerson; | | | |
|  | |  |  | | --- | --- | |  | Person.displayPerson; | | | |
|  | |  |  | | --- | --- | |  | person1.displayPerson(); | | | |
| **Question 3** | |  | 0 / 1 point | |

When is memory for a class allocated?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | when an object of the class is instantiated | | | |
|  | |  |  | | --- | --- | |  | when the class is declared | | | |
|  | |  |  | | --- | --- | |  | when function main is declared | | | |
|  | |  |  | | --- | --- | |  | when a constructor for class is declared | | | |
| **Question 4** | |  | 1 / 1 point | |

Members that are declared as public in a class can be accessed:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | inside and outside the class | | | |
|  | |  |  | | --- | --- | |  | only inside the class | | | |
|  | |  |  | | --- | --- | |  | only outside the class | | | |
| **Question 5** | |  | 1 / 1 point | |

True or false. Reference-type variables are initialized by default to the value null.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | true | | | |
|  | |  |  | | --- | --- | |  | false | | | |
| **Question 6** | |  | 1 / 1 point | |

Members that are declared as private in a class, can be accessed:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | inside and outside the class | | | |
|  | |  |  | | --- | --- | |  | only inside the class | | | |
|  | |  |  | | --- | --- | |  | only outside the class | | | |
| **Question 7** | |  | 1 / 1 point | |

What can be used to execute an application?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | A call by any method in a class to method main | | | |
|  | |  |  | | --- | --- | |  | Any class that contains public static void main ( )method | | | |
|  | |  |  | | --- | --- | |  | Any class | | | |
| **Question 8** | |  | 1 / 1 point | |

Multiple parameters are separated with what symbol?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | Dot separator | | | |
|  | |  |  | | --- | --- | |  | Comma. | | | |
|  | |  |  | | --- | --- | |  | Parentheses. | | | |
|  | |  |  | | --- | --- | |  | Braces. | | | |
| **Question 9** | |  | 1 / 1 point | |

Every Java application is composed of at least one:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | local variable | | | |
|  | |  |  | | --- | --- | |  | instance variable | | | |
|  | |  |  | | --- | --- | |  | public class declaration | | | |
|  | |  |  | | --- | --- | |  | imported class | | | |
| **Question 10** | |  | 1 / 1 point | |

True or false. Variables declared in the body of a particular method can be used in all methods of the class.

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | true | |
|  | |  |  | | --- | --- | |  | false | |

|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 1 / 1 point |

Given the following class definition:

public class A {

    int x;

    public A() {

        x = 4;

    }

    public void aMethod(){

        System.out.print(x);

    }

}

what would be printed by the following Java code?

A a = new A();

a.aMethod();

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | 0 | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
|  | |  |  | | --- | --- | |  | 4 | | | |
|  | |  |  | | --- | --- | |  | 44 | | | |
| **Question 2** | |  | 0 / 1 point | |

Which of the following is true about Objects in Java?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | Instance variables implement attributes, and methods implement behaviors. | | | |
|  | |  |  | | --- | --- | |  | Local variables implement attributes, and constructors implement behaviors. | | | |
|  | |  |  | | --- | --- | |  | Local variables implement attributes, and methods implement behaviors. | | | |
|  | |  |  | | --- | --- | |  | Instance variables implement attributes, and constructors implement behaviors. | | | |
| **Question 3** | |  | 0 / 1 point | |

Given the following class definition:

public class A {

    int x;

    public A(int y) {

        int x = y;

    }

    public aMethod(){

        System.out.print(x);

    }

}

    what would be printed by the following Java code?

          A a = new A(4);

    a.aMethod();

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | 0 | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
|  | |  |  | | --- | --- | |  | 4 | | | |
|  | |  |  | | --- | --- | |  | 44 | | | |
| **Question 4** | |  | 1 / 1 point | |

1.              Given the following statements, what is stored in newValue:

int i = 1;

int newValue;

int [] nums = {5, 10, 15, 20};

newValue = nums[i+1] + nums[i]/3;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | 0 | | | |
|  | |  |  | | --- | --- | |  | 11 | | | |
|  | |  |  | | --- | --- | |  | 15 | | | |
|  | |  |  | | --- | --- | |  | 18 | | | |
| **Question 5** | |  | 0 / 1 point | |

In the following Java code fragment:

**public** **class** A{

**int** x;

**public** **void** a() {

          x = 10;

       }

**public** **void** b() {

          a();

          /\*  state of x here? \*/

       }

    }

what is the best description of the state of the variable x at the indicated point?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | x will not exist | | | |
|  | |  |  | | --- | --- | |  | x will have value 10 | | | |
|  | |  |  | | --- | --- | |  | x will be available to be garbage collected | | | |
|  | |  |  | | --- | --- | |  | x will have no value | | | |
| **Question 6** | |  | 1 / 1 point | |

In the following Java code fragment:

public class MyClass {

    public void a() {

        int x;

        x = 17;

    }

    public void b() {

        a();

        /\* state of x here?\*/

    }

}

what is the best description of the state of the variable x at the indicated point?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | x will not exist | | | |
|  | |  |  | | --- | --- | |  | x will have value 17 | | | |
|  | |  |  | | --- | --- | |  | x will be available to be garbage collected | | | |
|  | |  |  | | --- | --- | |  | x will have no value | | | |
| **Question 7** | |  | 0 / 1 point | |

Given the following class definition:

public class A {

    int x;

    public A() {

        int x = 4;

    }

    public void aMethod(){

        System.out.print(x);

    }

}

what would be printed by the following Java code?

A a = new A();

a.aMethod();

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | 0 | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
|  | |  |  | | --- | --- | |  | 4 | | | |
|  | |  |  | | --- | --- | |  | 44 | | | |
| **Question 8** | |  | 1 / 1 point | |

In the following Java code fragment:

**public** **class** A{

**private** **int** n;

**private** String x;

**public** **static** **void** main(String []args) {

       A a = **new** A();

       System.***out***.println("n: " + a.n + " x: " + a.x);

   }

}

What will be the output ?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | n: 0 x: 0 | | | |
|  | |  |  | | --- | --- | |  | n: 0 x: null | | | |
|  | |  |  | | --- | --- | |  | n: null x: null | | | |
|  | |  |  | | --- | --- | |  | n: null x: 0 | | | |
| **Question 9** | |  | 0 / 1 point | |

Which of these is correct way of calling a constructor having no parameters of superclass A by subclass B?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | super(void); | | | |
|  | |  |  | | --- | --- | |  | superclass(); | | | |
|  | |  |  | | --- | --- | |  | super.A() | | | |
|  | |  |  | | --- | --- | |  | super(); | | | |
| **Question 10** | |  | 1 / 1 point | |

1.              Given String sName = "Joe Smith"; What is the value returned by sName.length()?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | |  |  | | --- | --- | |  | 7 | |
|  | | | |  |  | | --- | --- | |  | 8 | |
|  | | | |  |  | | --- | --- | |  | 9 | |
|  | | | |  |  | | --- | --- | |  | 10 | |
|  |  |
|  | |

|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 1 / 1 point |

Under the rules of inheritance, public methods in the superclass are

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | public in the subclass | | | |
|  | |  |  | | --- | --- | |  | protected in the subclass | | | |
|  | |  |  | | --- | --- | |  | accessible in the subclass | | | |
|  | |  |  | | --- | --- | |  | accessible through polymorphism only | | | |
| **Question 2** | |  | 1 / 1 point | |

A subclass (or derived class)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | incorporates data fields and methods from its superclass and can have additional fields and methods as well | | | |
|  | |  |  | | --- | --- | |  | incorporates data fields and methods from its superclass but cannot have additional fields and methods | | | |
|  | |  |  | | --- | --- | |  | incorporates data fields , but not methods, from its superclass and can have additional fields or methods as well | | | |
|  | |  |  | | --- | --- | |  | incorporates data fields, but not methods, from its superclass but cannot have additional fields or methods | | | |
| **Question 3** | |  | 1 / 1 point | |

Under the rules of inheritance, private data fields in the superclass are

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | not accessible in the subclass | | | |
|  | |  |  | | --- | --- | |  | protected in the subclass | | | |
|  | |  |  | | --- | --- | |  | public in the subclass | | | |
|  | |  |  | | --- | --- | |  | accessible through polymorphism only | | | |
| **Question 4** | |  | 1 / 1 point | |

With inheritance, a derived subclass object can directly access any

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | public or private superclass member | | | |
|  | |  |  | | --- | --- | |  | private superclass member | | | |
|  | |  |  | | --- | --- | |  | public superclass member (and protected subclass members if it's in the same package) | | | |
|  | |  |  | | --- | --- | |  | protected, public or private superclass member | | | |
| **Question 5** | |  | 1 / 1 point | |

The following prototype shows that a Cylinder subclass is derived from a superclass called Circle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | class Circle extends Cylinder | | | |
|  | |  |  | | --- | --- | |  | class Cylinder derived Circle | | | |
|  | |  |  | | --- | --- | |  | class Cylinder extends Circle | | | |
|  | |  |  | | --- | --- | |  | class Circle derived Cylinder | | | |
| **Question 6** | |  | 0 / 1 point | |

Inheritance refers to the ability to

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | |  |  | | --- | --- | |  | redefine how member methods of related classes operate | |
|  | | | |  |  | | --- | --- | |  | create new classes from existing ones | |
|  | | | |  |  | | --- | --- | |  | combine data fields and methods into one type | |
|  | | | |  |  | | --- | --- | |  | none of the above | |
|  |  |
|  | |

|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 1 / 1 point |

Which symbol is composition?

|  |  |  |  |
| --- | --- | --- | --- |
|  | Filled in diamond | | |
|  | None-triangular Arrow | | |
|  | Arrow with dotted line | | |
|  | Empty diamond | | |
|  | Triagular Arrow | | |
| **Question 2** | |  | 0 / 1 point | |

Which symbol is generalization ?

|  |  |  |  |
| --- | --- | --- | --- |
|  | Triagular Arrow | | |
|  | Arrow with dotted line | | |
|  | None-triangular Arrow | | |
|  | Filled in diamond | | |
|  | Empty diamond | | |
| **Question 3** | |  | 0 / 1 point | |

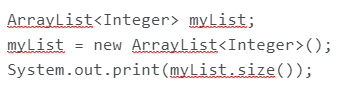
When we are defining a new interface, what happens if we leave out the abstract keyword in the method definitions?

|  |  |  |  |
| --- | --- | --- | --- |
|  | A compilation error would result because abstract methods are not allowed in an interface. | | |
|  | Because all methods in an interface are abstract, it would cause a compilation error. | | |
|  | Because the abstract keyword is for abstract classes, a compilation error would result. | | |
|  | Because all methods in an interface are abstract, it doesn’t matter whether or not we specify the abstract keyword. | | |
| **Question 4** | |  | 1 / 1 point | |

Which of the following statements about Abstract Classes and Interfaces are true?

|  |  |  |  |
| --- | --- | --- | --- |
|  | all attributes in an interface, if any, must be constants (i.e. public static final) | | |
|  | a class can extend only one abstract class, but a class can implement more than one interface | | |
|  | an interface doesn’t include implementation (method bodies) for any of its method definitions | | |
|  | all of the these | | |
| **Question 5** | |  | 1 / 1 point | |

What would be printed by executing the following Java code (assume that all required import statements are included)?



|  |  |  |  |
| --- | --- | --- | --- |
|  | 10 | | |
|  | Nothing, the code will not compile | | |
|  | 0 | | |
|  | -1 | | |
|  | None of these answers | | |
| **Question 6** | |  | 1 / 1 point | |

What Java keyword is used to inherit an interface?

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | overrides |
|  | | | overloads |
|  | | | inherits |
|  | | | extends |
|  | | | implements |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 1 / 1 point |

How do you print all the  elements of the following ArrayList?

ArrayList<String> cars = new ArrayList<String>();

|  |  |  |  |
| --- | --- | --- | --- |
|  | for (String str : cars) {          System.out.println(str);       } | | |
|  | for (String cars : str) {          System.out.println(str);       } | | |
|  | for (int i = 0; i < cars.size(); i++) {          System.out.println(cars[i]);      } | | |
|  | for (int i = 0; i < cars.length(); i++) {          System.out.println(cars.get(i));      } | | |
| **Question 2** | |  | 1 / 1 point | |

What is the syntax to create an ArrayList of Strings named cars?

|  |  |  |  |
| --- | --- | --- | --- |
|  | ArrayList cars = new ArrayList<String>(); | | |
|  | ArrayList<String> cars = new ArrayList(); | | |
|  | <String> cars = new ArrayList<String>(); | | |
|  | ArrayList<String> cars = new ArrayList<String>(); | | |
| **Question 3** | |  | 1 / 1 point | |

How do you insert an item into the following ArrayList?

ArrayList<String> cars = new ArrayList<String>();

|  |  |  |  |
| --- | --- | --- | --- |
|  | cars.add("Lexus"); | | |
|  | cars.include("Lexus"); | | |
|  | cars.insert("Lexus"); | | |
|  | cars.addItem("Lexus"); | | |
| **Question 4** | |  | 1 / 1 point | |

How do you insert an item at the first position of the following ArrayList?

ArrayList<String> cars = new ArrayList<String>();

|  |  |  |  |
| --- | --- | --- | --- |
|  | cars[0]= "Toyota"); | | |
|  | cars.insert(0, "Toyota"); | | |
|  | cars.addItem(0, "Toyota"); | | |
|  | cars.add(0, "Toyota"); | | |
| **Question 5** | |  | 1 / 1 point | |

Which statement is true for an ArrayList?

|  |  |  |  |
| --- | --- | --- | --- |
|  | Values or references are stored in contiguous locations | | |
|  | References are stored in contiguous locations | | |
| **Question 6** | |  | 1 / 1 point | |

The size of an ArrayList is

|  |  |  |  |
| --- | --- | --- | --- |
|  | limited | | |
|  | dynamic | | |
|  | fixed | | |
|  | constant | | |
| **Question 1** | |  | 1 / 1 point |

Use enums when you have values that you know aren't going to change, like month names, days of the week, colors, card suits, etc.

|  |  |  |
| --- | --- | --- |
|  |  | True |
|  |  | False |

|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 1 / 1 point |

The purpose of exception handling (using try...catch) is to handle common errors in your program.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | a) True | | |
|  |  | b) False | | |
| **Question 2** | | |  | 1 / 1 point | |

The exception thrown when the program executes a division where the denominator is 0 is:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | IOException | | | |
|  | |  |  | | --- | --- | |  | ArithmeticException | | | |
|  | |  |  | | --- | --- | |  | InputMismatchException | | | |
|  | |  |  | | --- | --- | |  | none of the above | | | |
| **Question 3** | |  | 1 / 1 point | |

Which of the following are examples of common exceptions?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | arithmetic overflow | | | |
|  | |  |  | | --- | --- | |  | division by zero | | | |
|  | |  |  | | --- | --- | |  | memory exhaustion | | | |
|  | |  |  | | --- | --- | |  | array index in bounds | | | |
| **Question 4** | |  | 1 / 1 point | |

What happens if multiple catch blocks match the type of thrown object?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | The last one that matches the exception is executed. | | | |
|  | |  |  | | --- | --- | |  | The current program thread stops executing | | | |
|  | |  |  | | --- | --- | |  | The first catch block after the try that matches the exception is executed. | | | |
|  | |  |  | | --- | --- | |  | none of the above | | | |
| **Question 5** | |  | 1 / 1 point | |

The difference between a checked exception and unchecked exception is:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | The compilor verifies that all checked exceptions thrown by each method are caught in the calling code (or declared in a throw clause). | | | |
|  | |  |  | | --- | --- | |  | The JVM verifies that all checked exceptions thrown by each method are caught in the calling code (or declared in a throw clause) | | | |
|  | |  |  | | --- | --- | |  | Java catches all checked exceptions automatically if the programmer has not done so. | | | |
|  | |  |  | | --- | --- | |  | none of the above | | | |
| **Question 6** | |  | 1 / 1 point | |

The exception thrown by the Scanner class when executing method nextInt, and invalid chars are entered for the int is:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | ArithmeticException | | | |
|  | |  |  | | --- | --- | |  | IndexOutOfBoundsException | | | |
|  | |  |  | | --- | --- | |  | InputMismatchException | | | |
|  | |  |  | | --- | --- | |  | IOException | | | |
| **Question 7** | |  | 1 / 1 point | |

What happens if no catch handler matches the tpe of a thrown object?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | If there is a finally block, it is executed and a search for the match is continued in the next enclosing try statement. (If no match is found, the program thread terminates early). | | | |
|  | |  |  | | --- | --- | |  | The program thread terminates early. | | | |
|  | |  |  | | --- | --- | |  | The exception is ignored. | | | |
|  | |  |  | | --- | --- | |  | none of the above. | | | |
| **Question 8** | |  | 1 / 1 point | |

Does declaring a catch of a superclass exception catch all of the subclass exceptions of that superclass?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 1) True | | |
|  |  | 2) False | | |
| **Question 9** | | |  | 1 / 1 point | |

When a try block is executed but no exceptions are thrown, where does control pass when the try block has completed execution?

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | |  |  | | --- | --- | |  | The last catch block. | | | |
|  | | | |  |  | | --- | --- | |  | The finally block, if there is one, otherwise to the code after the last catch block. | | | |
|  | | | |  |  | | --- | --- | |  | The code after the last catch block. | | | |
|  | | | |  |  | | --- | --- | |  | none of the above. | | | |
|  |  |
|  | |
| **Question 1** | | | |  | 1 / 1 point |

How do you print all the  elements of the following ArrayList?

ArrayList<String> cars = new ArrayList<String>();

|  |  |  |  |
| --- | --- | --- | --- |
|  | for (String str : cars) {          System.out.println(str);       } | | |
|  | for (String cars : str) {          System.out.println(str);       } | | |
|  | for (int i = 0; i < cars.size(); i++) {          System.out.println(cars[i]);      } | | |
|  | for (int i = 0; i < cars.length(); i++) {          System.out.println(cars.get(i));      } | | |
| **Question 2** | |  | 1 / 1 point | |

What is the syntax to create an ArrayList of Strings named cars?

|  |  |  |  |
| --- | --- | --- | --- |
|  | ArrayList cars = new ArrayList<String>(); | | |
|  | ArrayList<String> cars = new ArrayList(); | | |
|  | <String> cars = new ArrayList<String>(); | | |
|  | ArrayList<String> cars = new ArrayList<String>(); | | |
| **Question 3** | |  | 1 / 1 point | |

How do you insert an item into the following ArrayList?

ArrayList<String> cars = new ArrayList<String>();

|  |  |  |  |
| --- | --- | --- | --- |
|  | cars.add("Lexus"); | | |
|  | cars.include("Lexus"); | | |
|  | cars.insert("Lexus"); | | |
|  | cars.addItem("Lexus"); | | |
| **Question 4** | |  | 1 / 1 point | |

How do you insert an item at the first position of the following ArrayList?

ArrayList<String> cars = new ArrayList<String>();

|  |  |  |  |
| --- | --- | --- | --- |
|  | cars[0]= "Toyota"); | | |
|  | cars.insert(0, "Toyota"); | | |
|  | cars.addItem(0, "Toyota"); | | |
|  | cars.add(0, "Toyota"); | | |
| **Question 5** | |  | 1 / 1 point | |

Which statement is true for an ArrayList?

|  |  |  |  |
| --- | --- | --- | --- |
|  | Values or references are stored in contiguous locations | | |
|  | References are stored in contiguous locations | | |
| **Question 6** | |  | 1 / 1 point | |

The size of an ArrayList is

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | limited |
|  | | | dynamic |
|  | | | fixed |
|  | | | constant |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 1 / 1 point |

Exception class 'ArithmeticException' is in which package?

|  |  |  |  |
| --- | --- | --- | --- |
|  | java.util | | |
|  | java.io | | |
|  | java.lang | | |
|  | java.exception | | |
| **Question 2** | |  | 1 / 1 point | |

The class at the top of exception class hierarchy is

|  |  |  |  |
| --- | --- | --- | --- |
|  | ArithmeticException | | |
|  | Error | | |
|  | Throwable | | |
|  | Exception | | |
| **Question 3** | |  | 0 / 1 point | |

Which of the following statements is true?

A. Error is a subclass of Throwable  
B. Error is a subclass of Exception  
C. Error indicates abnormal situations that should not be caught   
D. ArithmeticException is an Error

|  |  |  |  |
| --- | --- | --- | --- |
|  | B & C | | |
|  | A & B | | |
|  | B & D | | |
|  | A & C | | |
| **Question 4** | |  | 1 / 1 point | |

Which of the following is an unchecked exception?

|  |  |  |  |
| --- | --- | --- | --- |
|  | ArithmeticException | | |
|  | FileNotFoundException | | |
|  | IOException | | |
|  | All of these | | |
| **Question 5** | |  | 1 / 1 point | |

Which keyword is used to explicitly throw an exception?

|  |  |  |  |
| --- | --- | --- | --- |
|  | throw | | |
|  | finally | | |
|  | catch | | |
|  | try | | |
| **Question 6** | |  | 1 / 1 point | |

Which exception will be thrown when there is a division by zero is executed in the code?

|  |  |  |  |
| --- | --- | --- | --- |
|  | NullPointerException | | |
|  | InputMismatchException | | |
|  | ArithmeticException | | |
|  | NumberFormatException | | |
| **Question 7** | |  | 1 / 1 point | |

Which of the following is a checked exception?

|  |  |  |  |
| --- | --- | --- | --- |
|  | NullPointerException | | |
|  | None of these | | |
|  | IOException | | |
|  | ArithmeticException | | |
| **Question 8** | |  | 1 / 1 point | |

Which block will be executed irrespective of whether an exception is caught or not?

|  |  |
| --- | --- |
|  | finally |
|  | catch |
|  | throws |
|  | throw |

|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 1 / 1 point |

The following code will execute successfully: (assuming the file payables.txt contains as it's first data 150.00):

Scanner inPayable = new Scanner (new File ("payables.txt"));

Payables record = (PayablesRecord) inPayable.readObject();

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 1) True | | |
|  |  | 2) False | | |
| **Question 2** | | |  | 1 / 1 point | |

The statement(s) that reads a record from the file "trans.ser" into an object of class TransactionRecord (using the ObjectInputStream variable inTrans which has been declared successfully already) is:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | TransactionRecord trans = inTrans.readObject(); | | | |
|  | |  |  | | --- | --- | |  | TransactionRecord trans = (TransactionRecord) inTrans.readObject(); | | | |
|  | |  |  | | --- | --- | |  | TransactionRecord trans = (TransactionRecord) inTrans.writeObject(); | | | |
|  | |  |  | | --- | --- | |  | none of the above. | | | |
| **Question 3** | |  | 1 / 1 point | |

When using object serialization to save and restore data in your program to a file, every class in your project that models data to be saved/restored must implement serializable.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 1) True | | |
|  |  | 2) False | | |
| **Question 4** | | |  | 1 / 1 point | |

The statement that opens file "trans.ser" for input using ObjectInputStream variable inTrans to wrap a FileInputStream object is:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | ObjectInputStream inTrans = ObjectInputStream ("trans.ser"); | | | |
|  | |  |  | | --- | --- | |  | ObjectInputStream inTrans = new ObjectInputStream ("trans.ser"); | | | |
|  | |  |  | | --- | --- | |  | ObjectInputStream inTrans = new ObjectInputStream (new FileInputStream("trans.ser")); | | | |
|  | |  |  | | --- | --- | |  | none of the above | | | |
| **Question 5** | |  | 1 / 1 point | |

If an absolute path of a file is NOT specified when declaring a file object and connecting to that file to read from it, the file must exist be in the project directory.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 1) True | | |
|  |  | 2) False | | |
| **Question 6** | | |  | 1 / 1 point | |

An absolute path contains all the directories, starting with the root directory, that lead to a specific file or directory.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 1) True | | |
|  |  | 2) False | | |
| **Question 7** | | |  | 1 / 1 point | |

When reading data from a file, if you wish to read data in the file multiple times, the file must be closed and reopened to read from the beginning of the file.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 1) True | | |
|  |  | 2) False | | |
| **Question 8** | | |  | 1 / 1 point | |

The statement that outputs a record of type TransactionRecord (stored in object transactionRecord) to the file "newTrans.ser" (which has been connected to the object outTrans of type ObjectOutputStream successfully already) is:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | outTrans.writeObject(); | | |
|  | |  |  | | --- | --- | |  | outTrans = writeObject(); | | |
|  | |  |  | | --- | --- | |  | outTrans.writeObject("newTrans.ser"); | | |
|  | |  |  | | --- | --- | |  | none of the above. | | |
| [View Feedback](javascript://) | |

|  |  |  |
| --- | --- | --- |
| **Question 9** |  | 1 / 1 point |

The statement that opens file "newTrans.ser" for output using ObjectOutputStream variable outTrans to wrap a FileOutputStream object is:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | ObjectOutputStream outTrans = new ObjectOutputStream (new FileOutputStream ("newtrans.ser")); | | | |
|  | |  |  | | --- | --- | |  | ObjectOutputStream outTrans = new ObjectInputStream (new FileInputStream ("newtrans.ser")); | | | |
|  | |  |  | | --- | --- | |  | ObjectInputStream outTrans = new ObjectOutputStream (new FileOutputStream ("newtrans.ser")); | | | |
|  | |  |  | | --- | --- | |  | none of the above | | | |
| **Question 10** | |  | 1 / 1 point | |

The following code will execute successfully: (assuming the file payables.txt contains as it's first data 150.00):

Scanner inPayable = new Scanner (new File ("payables.txt"));

double amount = inPayable.nextDouble();

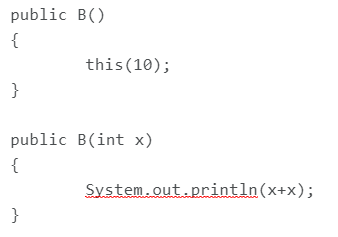
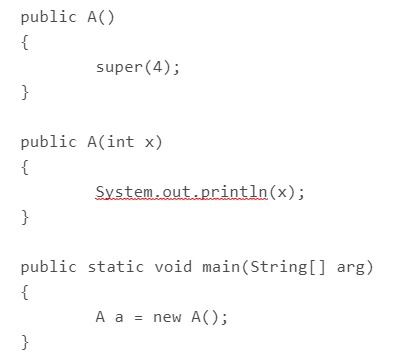
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 1) True | | |
|  |  | 2) False | | |
| **Question 11** | | |  | 1 / 1 point | |

Binary files are human readable in a text editor.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | | 1) True | | |
|  |  | | 2) False | | |
|  |  |
|  | |
| **Question 1** | | | |  | 1 / 1 point |

The default constructor of a subclass always calls the default constructor of it's superclass

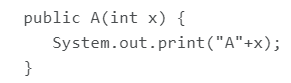
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | True | | |
|  |  | False | | |
| **Question 2** | | |  | 1 / 1 point | |

You have a class named B with the following in it:   
  
  
You have a class named A that extends class B, and has the following in it:  


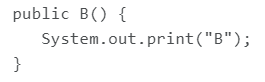
What will be the output?

|  |  |  |  |
| --- | --- | --- | --- |
|  | 20 | | |
|  | 8 | | |
|  | 4 | | |
|  | 10 | | |
|  | Compile error | | |
| **Question 3** | |  | 1 / 1 point | |

You have a class named A with the following in it:



You have a class named B that extends class A, and has the following in it:

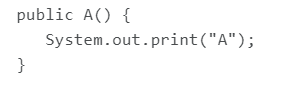


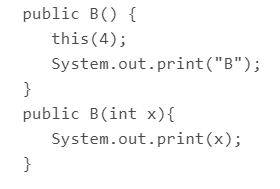
What is printed out by the following Java statement:

A a = new B();

|  |  |  |  |
| --- | --- | --- | --- |
|  | A0 | | |
|  | none of these | | |
|  | B | | |
|  | A0B | | |
|  | BA0 | | |
| **Question 4** | |  | 1 / 1 point | |

You have a class named A with the following in it:



You have a class named B that extends class A, and has the following in it:  


What is printed out by the following Java statement:

B b = new B();

|  |  |  |  |
| --- | --- | --- | --- |
|  | AB | | |
|  | A4B | | |
|  | 4AB | | |
|  | None of these | | |
|  | AB4 | | |
| **Question 5** | |  | 0 / 1 point | |

Can a constructor call another constructor of the same class?

|  |  |  |  |
| --- | --- | --- | --- |
|  | Never | | |
| Correct Answer | Yes, but the call must be the first statement of the constructor | | |
|  | Yes, but the call must be the last statement of the constructor | | |
| Incorrect Response | Yes, and it can be anywhere in the constructor | | |
| **Question 6** | |  | 1 / 1 point | |

Which of these is the correct way of calling a constructor having no parameters, of superclass A by subclass B?

|  |  |  |  |
| --- | --- | --- | --- |
|  | super(); | | |
|  | super(void); | | |
|  | super.A(); | | |
|  | superclass.(); | | |
| **Question 7** | |  | 0 / 1 point | |

When working with a Scanner in Java, its next method will return

|  |  |  |  |
| --- | --- | --- | --- |
| Correct Answer | The next word of input | | |
|  | The next integer value | | |
| Incorrect Response | The next line of input | | |
|  | The next Unicode character | | |
|  | None of these answers | | |
| **Question 8** | |  | 1 / 1 point | |

Which method can be used to find the length of a string?

|  |  |  |  |
| --- | --- | --- | --- |
|  | size() | | |
|  | length() | | |
|  | getLength() | | |
|  | len() | | |
|  | getSize() | | |
| **Question 9** | |  | 0 / 1 point | |

Which keyword is used to import a package from the Java API library?

|  |  |  |  |
| --- | --- | --- | --- |
| Correct Answer | import | | |
|  | getlib | | |
|  | lib | | |
| Incorrect Response | package | | |
| **Question 10** | |  | 1 / 1 point | |

Which of the following is true about **Objects** in Java?

|  |  |  |  |
| --- | --- | --- | --- |
|  | Instance variables implement attributes, and constructors implement behaviors. | | |
|  | Local variables implement attributes, and constructors implement behaviors. | | |
|  | Local variables implement attributes, and methods implement behaviors. | | |
|  | Instance variables implement attributes, and methods implement behaviors. | | |
|  | None of these answers. | | |
| **Question 11** | |  | 1 / 1 point | |

Which of the following statements is **true**:

|  |  |  |  |
| --- | --- | --- | --- |
|  | A List can contain both primitives and reference types | | |
|  | Polymorphism refers to the bundling of data with the methods that operate on that data | | |
|  | A package is intended to group related objects | | |
|  | A multidimensional array cannot have more than three dimensions | | |
|  | A final variable cannot be changed | | |
| **Question 12** | |  | 0 / 1 point | |

Which of the following statements about toString() method of a user-defined class is false:

|  |  |  |  |
| --- | --- | --- | --- |
| Correct Answer | can be overloaded if multiple String representations are required | | |
|  | overrides the toString() method of the Object class | | |
| Incorrect Response | If you print an object, java compiler internally invokes it | | |
|  | returns the string representation of the object | | |
|  | the default behavior is to return a string containing the class name | | |
| **Question 13** | |  | 0 / 1 point | |

When should a program *explicitly* use the **this** reference?

|  |  |  |  |
| --- | --- | --- | --- |
| Correct Answer | Accessing a field that has the same name as a local variable. | | |
| Incorrect Response | Accessing a local variable | | |
|  | Accessing a public variable | | |
|  | Accessing a private variable. | | |
| **Question 14** | |  | 1 / 1 point | |

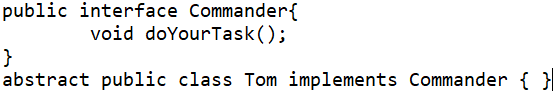
Which of these declares an abstract method in an existing abstract Java class?

|  |  |  |  |
| --- | --- | --- | --- |
|  | public void myMethod() { }; | | |
|  | public abstract void myMethod(); | | |
|  | public abstract void  myMethod() { }; | | |
|  | public abstract myMethod(); | | |
|  | public void abstract myMethod(); | | |
| **Question 15** | |  | 1 / 1 point | |

Choose the class definition that defines a legal abstract class.

|  |  |  |  |
| --- | --- | --- | --- |
|  | class D { abstract void incomplete() { }; } | | |
|  | abstract class D { abstract void incomplete(); } | | |
|  | public class abstract D { abstract void incomplete(); } | | |
|  | class D { abstract void incomplete(); } | | |
| **Question 16** | |  | 0 / 1 point | |

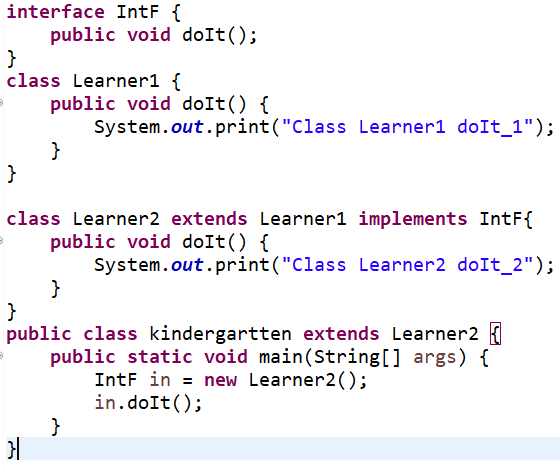
Given the following piece of code:



which of the the statements below is correct?

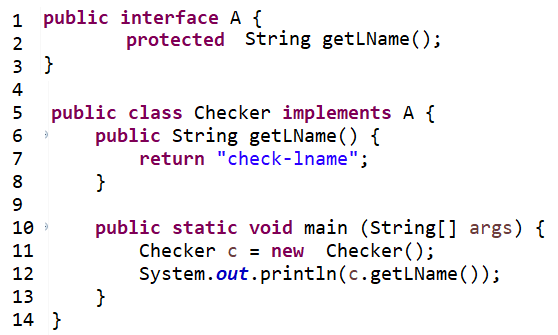
|  |  |  |  |
| --- | --- | --- | --- |
| Correct Answer | The program will compile without errors. | | |
| Incorrect Response | The code won't compile since class Tom must implement method doYourTask() from interface Commander. | | |
|  | The program will not compile since method doYourTask() in interface Commander must be defined as abstract. | | |
|  | The program will not compile since declaration of class Tom uses keyword implements instead of extends. | | |
| **Question 17** | |  | 1 / 1 point | |

What is the output of the program below?



|  |  |  |  |
| --- | --- | --- | --- |
|  | Class Learner2 doIt\_2 | | |
|  | gets compiled fully but prints nothing | | |
|  | none of these | | |
|  | Class Learner1 doIt\_1 | | |
|  | Compilation error | | |
| **Question 18** | |  | 0 / 1 point | |

What statement is true about the outcome of running the program below ?



|  |  |  |  |
| --- | --- | --- | --- |
|  | compilation fails due to an error in line 1 | | |
|  | compilation is successful but prints nothing | | |
|  | None of these | | |
| Correct Answer | compilation fails due to an error in line 2 | | |
| Incorrect Response | prints checklname when run | | |
| **Question 19** | |  | 1 / 1 point | |

Which of the following statements about Abstract Classes and Interfaces are true?

|  |  |  |  |
| --- | --- | --- | --- |
|  | all of these | | |
|  | a class can extend only one abstract class, but a class can implement more than one interface | | |
|  | an abstract class doesn’t include implementation (method bodies) for any of its method definitions | | |
|  | all attributes in an abstract class, if any, must be constants (i.e. public static final) | | |
| **Question 20** | |  | 0 / 1 point | |

Which of the following statements is **true:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | an abstract class must contain at least one abstract method | | |
| Correct Answer | an interface can contain constants | | |
| Incorrect Response | an interface can have protected instance variables | | |
|  | an abstract class cannot extend a concrete class | | |
| **Question 21** | |  | 0 / 1 point | |

Which of the following is **false**:

|  |  |  |  |
| --- | --- | --- | --- |
| Correct Answer | You cannot extend a concrete class to an abstract class | | |
|  | An abstract class can have any number of constructors | | |
| Incorrect Response | A class declared with abstract keyword and have zero or more abstract methods is known as abstract class. | | |
|  | To use, you need to extend an abstract classes to a concrete class. | | |
| **Question 22** | |  | 1 / 1 point | |

Which of the following is **false**:

|  |  |  |  |
| --- | --- | --- | --- |
|  | A Java class can implement multiple interfaces but it can extend only one abstract class. | | |
|  | An interface can extend another Java interface only, an abstract class can extend another Java class and implement multiple Java interfaces. | | |
|  | Variables declared in a Java interface is by default final. An abstract class may contain non-final variables. | | |
|  | A Java interface are implicitly abstract and cannot have implementations. A Java abstract class can have instance methods that implements a default behavior. | | |
|  | Members of an abstract class are public by default. An interface can have private, protected, etc.. | | |
| **Question 23** | |  | 1 / 1 point | |

Which of the following is **false**:

|  |  |  |  |
| --- | --- | --- | --- |
|  | A Java class can implement multiple interfaces but it can extend only one abstract class. | | |
|  | A Java interface are implicitly abstract and cannot have implementations. A Java abstract class can have instance methods that implements a default behavior. | | |
|  | Variables declared in a Java abstract class is by default final. An interface may contain non-final variables. | | |
|  | An interface can extend another Java interface only, an abstract class can extend another Java class and implement multiple Java interfaces. | | |
|  | Members of a Java interface are public by default. A Java abstract class can have private, protected, etc.. | | |
| **Question 24** | |  | 1 / 1 point | |

Which of the following statements about Abstract Classes and Interfaces are true?

|  |  |  |  |
| --- | --- | --- | --- |
|  | a class can extend only one abstract class, but a class can implement more than one interface | | |
|  | an interface doesn’t include implementation (method bodies) for any of its method definitions | | |
|  | all of the these | | |
|  | all attributes in an interface, if any, must be constants (i.e. public static final) | | |
| **Question 25** | |  | 0 / 1 point | |

Which of the following is **false**:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Members of a Java interface are public by default. A Java abstract class can have private, protected, etc.. | | |
| Incorrect Response | An interface can extend another Java interface only, an abstract class can extend another Java class and implement multiple Java interfaces. | | |
|  | Variables declared in a Java interface is by default final. An abstract class may contain non-final variables. | | |
| Correct Answer | A Java abstract class is implicitly abstract and cannot have implementations. A Java interface can have instance methods that implements a default behavior. | | |
|  | A Java class can implement multiple interfaces but it can extend only one abstract class. | | |
| **Question 26** | |  | 1 / 1 point | |

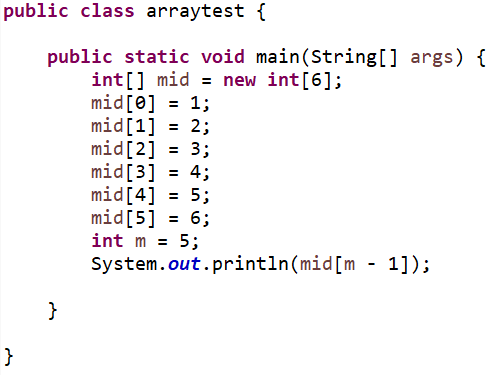
Java Interfaces are useful because:

|  |  |  |  |
| --- | --- | --- | --- |
|  | they form a contract specifying what behaviour is provided by a class that implements the interface without forcing that behaviour to be implemented in a certain way | | |
|  | they can help define an Application Programming Interface (API) | | |
|  | all of these | | |
|  | they can help coordinate between two separate groups of programmers when one group needs to use methods that the other group implements | | |
| **Question 27** | |  | 1 / 1 point | |

Which of the following is **true** of an abstract class in Java?

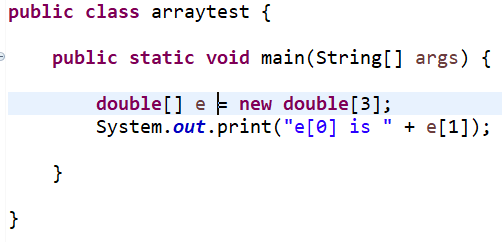
|  |  |  |  |
| --- | --- | --- | --- |
|  | An abstract class can be used as the superclass in polymorphic code. | | |
|  | An abstract class can only define abstract methods and cannot implement methods. | | |
|  | None of these answers | | |
|  | An abstract class is used to instantiate abstract objects. | | |
|  | An abstract class cannot be extended. | | |
| **Question 28** | |  | 1 / 1 point | |

What is the output of the code fragment shown below?



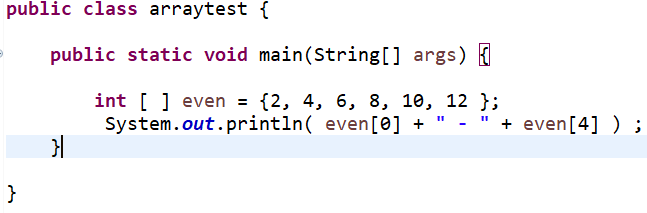
|  |  |  |  |
| --- | --- | --- | --- |
|  | 2 | | |
|  | 4 | | |
|  | 5 | | |
|  | 3 | | |
| **Question 29** | |  | 1 / 1 point | |

What is does the program below output?



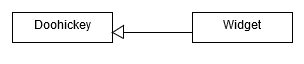
|  |  |  |  |
| --- | --- | --- | --- |
|  | a compile time error since an incorrect array size was specified in the array declaration | | |
|  | a runtime error caused by e [1] in the output statement | | |
|  | a runtime error caused by failing to initialize array elements | | |
|  | The program compiles and runs fine and displaying e[0] is 0.0 as output | | |
| **Question 30** | |  | 1 / 1 point | |

What is the output of program?



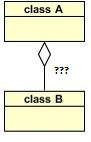
|  |  |  |  |
| --- | --- | --- | --- |
|  | 2 -11 | | |
|  | 2 - 10 | | |
|  | 2 - 6 | | |
|  | 2 - 8 | | |
| **Question 31** | |  | 0 / 1 point | |

What does the following mean?



|  |  |  |  |
| --- | --- | --- | --- |
|  | A Widget has a Doohickey | | |
| Incorrect Response | A Doohickey has a Widget | | |
|  | A Doohickey is a Widget | | |
|  | A Widget use a Doohickey | | |
| Correct Answer | A Widget is a Doohickey | | |
| **Question 32** | |  | 1 / 1 point | |

What is the best word for the relationship?

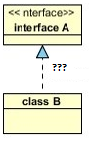


|  |  |  |  |
| --- | --- | --- | --- |
|  | owns | | |
|  | is-a | | |
|  | uses | | |
|  | realizes | | |
|  | contains | | |
| **Question 33** | |  | 1 / 1 point | |

Which symbol is realization ?

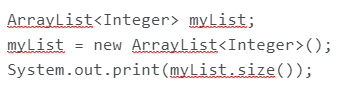
|  |  |  |  |
| --- | --- | --- | --- |
|  | Triagular Arrow | | |
|  | None-triangular Arrow | | |
|  | Filled in diamond | | |
|  | Arrow with dotted line | | |
|  | Empty diamond | | |
| **Question 34** | |  | 1 / 1 point | |

What is the best word for the relationship?



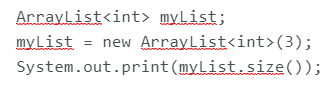
|  |  |  |  |
| --- | --- | --- | --- |
|  | realizes | | |
|  | owns | | |
|  | uses | | |
|  | contains | | |
|  | is-a | | |
| **Question 35** | |  | 1 / 1 point | |

What would be printed by executing the following Java code (assume that all required import statements are included)?



|  |  |  |  |
| --- | --- | --- | --- |
|  | 10 | | |
|  | Nothing, the code will not compile | | |
|  | 0 | | |
|  | -1 | | |
|  | None of these answers | | |
| **Question 36** | |  | 1 / 1 point | |

What would be printed by executing the following Java code (assume that all required import statements are included)?



|  |  |  |  |
| --- | --- | --- | --- |
|  | None of these answers | | |
|  | Nothing, the code will not compile | | |
|  | -1 | | |
|  | 0 | | |
|  | 3 | | |
| **Question 37** | |  | 2 / 4 points | |

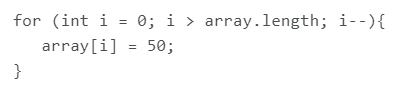
Place the following access levels in order from most restrictive to least restrictive:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | \_\_1\_\_ |  | private | | |
|  | \_\_4\_\_ |  | public | | |
| Incorrect Response | \_\_2\_\_ | **(3)** | protected | | |
| Incorrect Response | \_\_3\_\_ | **(2)** | default (no modifier) | | |
| **Question 38** | | | |  | 3 / 4 points |

Match "Subclass Diff Pkg." member to it's visibility

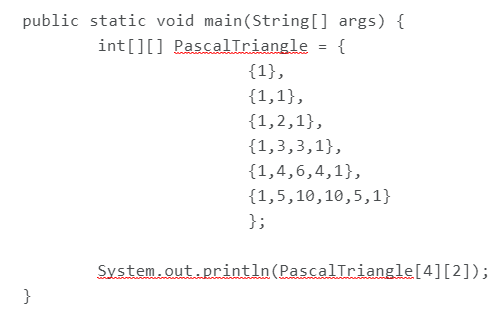
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  | \_\_1\_\_ |  | public | |  | \_\_1\_\_ |  | protected | |  | \_\_2\_\_ |  | private | | Incorrect Response | \_\_1\_\_ | **(2)** | default (no modifier) | |  | |  |  | | --- | --- | | **1**. | Visible | | **2**. | Not visible | | | |
| **Question 39** | | |  | 1 / 1 point |

You have an array named array of type int and of size 10. What is true of the following Java code?



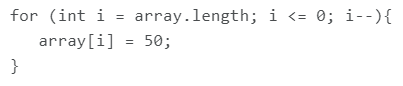
|  |  |  |  |
| --- | --- | --- | --- |
|  | it would result in an IndexOutOfBoundsException being thrown | | |
|  | it would assign all of the elements of array the value 0 | | |
|  | it would result in a compile-time error | | |
|  | it would generate a runtime error | | |
|  | it would assign all of the elements of array the value 50 | | |
| **Question 40** | |  | 1 / 1 point | |

What is the output of the following?:



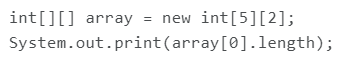
|  |  |  |  |
| --- | --- | --- | --- |
|  | nothing, the code won’t compile | | |
|  | 3 | | |
|  | runtime exception | | |
|  | 6 | | |
|  | 4 | | |
| **Question 41** | |  | 0 / 1 point | |

You have an array named array of type int and of size 10. What is true of the following Java code?



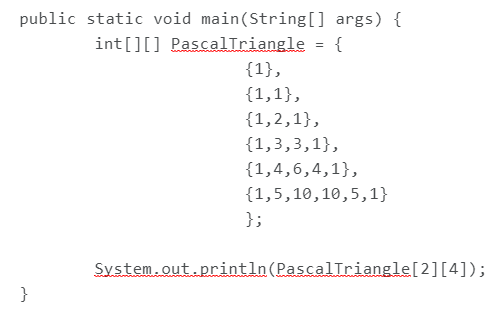
|  |  |  |  |
| --- | --- | --- | --- |
|  | it would result in a compile-time error | | |
|  | it would generate a runtime error | | |
|  | it would assign all of the elements of array the value 50 | | |
| Incorrect Response | it would result in an IndexOutOfBoundsException being thrown | | |
| Correct Answer | it would assign all of the elements of array the value 0 | | |
| **Question 42** | |  | 1 / 1 point | |

What is result of the following Java code?



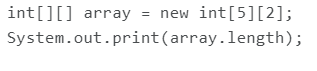
|  |  |  |  |
| --- | --- | --- | --- |
|  | 2 | | |
|  | 5 | | |
|  | 10 | | |
|  | 0 | | |
|  | None of the above | | |
| **Question 43** | |  | 1 / 1 point | |

What is the output of the following?:



|  |  |  |  |
| --- | --- | --- | --- |
|  | runtime exception | | |
|  | nothing, the code won’t compile | | |
|  | 4 | | |
|  | 6 | | |
|  | 3 | | |
| **Question 44** | |  | 1 / 1 point | |

What is result of the following Java code?



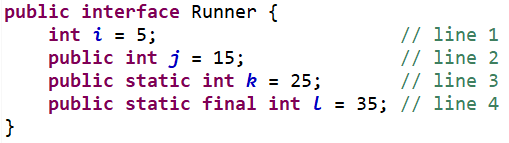
|  |  |  |  |
| --- | --- | --- | --- |
|  | | | 2 |
|  | | | 5 |
|  | | | 10 |
|  | | | 0 |
|  | | | None of the above |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 0 / 1 point |

Which of these is a correct interface?

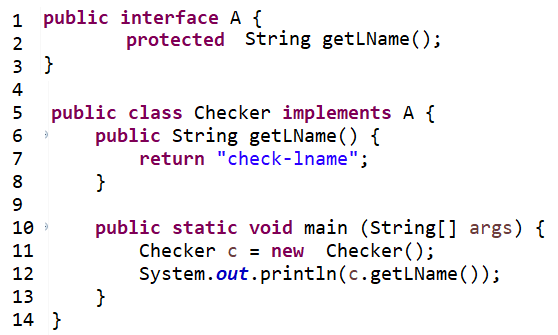
|  |  |
| --- | --- |
|  | abstract interface B { print (); } |
|  | abstract interface B { abstract void print (); { } } |
| Correct Answer | interface B { void print (); } |
| Incorrect Response | interface B { void print () { } } |
| **Question 2** | |  | 1 / 1 point |

Which of the lines below will give compilation error?



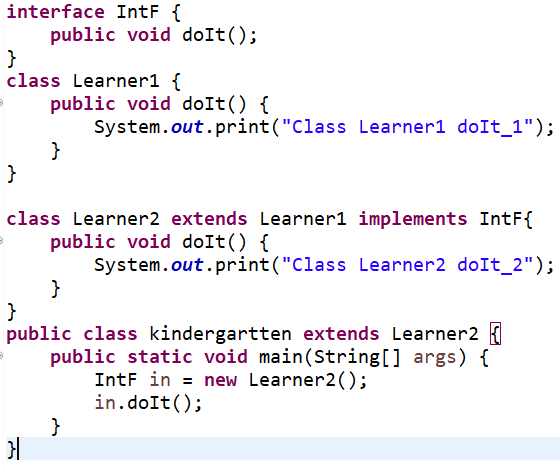
|  |  |
| --- | --- |
|  | 2 |
|  | 3 |
|  | 1 |
|  | 4 |
|  | None |
| **Question 3** | |  | 0 / 1 point |

What statement is true about the outcome of running the program below ?



|  |  |
| --- | --- |
| Correct Answer | compilation fails due to an error in line 2 |
|  | compilation fails due to an error in line 1 |
|  | None of these |
| Incorrect Response | prints checklname when run |
|  | compilation is successful but prints nothing |
| **Question 4** | |  | 1 / 1 point |

What is the output of the program below?



|  |  |
| --- | --- |
|  | gets compiled fully but prints nothing |
|  | Class Learner1 doIt\_1 |
|  | none of these |
|  | Compilation error |
|  | Class Learner2 doIt\_2 |
| **Question 5** | |  | 0 / 1 point |

Suppose B is an abstract class, C is a concrete subclass of B, and both B and C have a no-arg constructor. Which of the following is correct?

1. B  b = new B();

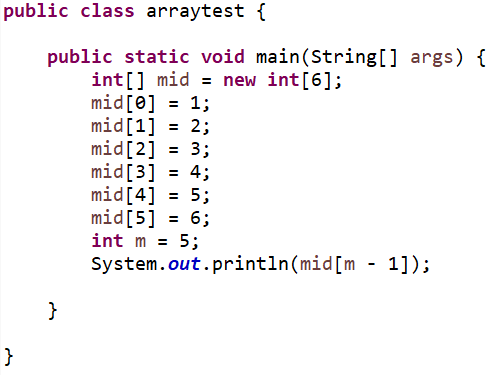
2. B  b = new C ();

3. C  c = new B();

4. C  c = new C();

|  |  |
| --- | --- |
| Correct Answer | 2 and 4 |
|  | 2 and 3 |
|  | 1 and 3 |
| Incorrect Response | 3 and 4 |
| **Question 6** | |  | 1 / 1 point |

What is the output of the code fragment shown below?

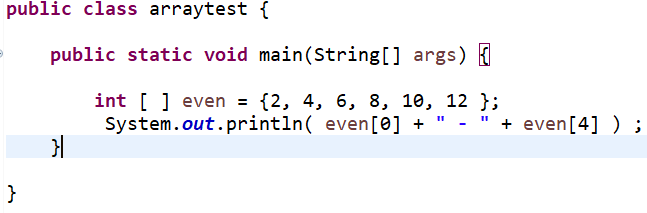


|  |  |
| --- | --- |
|  | 4 |
|  | 5 |
|  | 3 |
|  | 2 |
| **Question 7** | |  | 1 / 1 point |

What element is at position multiArr [2][3]?  
int [ ] [ ] multiArr =  {  {2, 3, 4, 5},  {6, 7, 8, 9},  {10, 11, 12, 13},  {14, 15, 16, 17}  };

|  |  |
| --- | --- |
|  | 8 |
|  | 12 |
|  | 13 |
|  | 16 |
| **Question 8** | |  | 1 / 1 point |

What is the output of program?



|  |  |
| --- | --- |
|  | 2 -11 |
|  | 2 - 10 |
|  | 2 - 6 |
|  | 2 - 8 |
| **Question 9** | |  | 0 / 1 point |

Which of the following statements about Abstract Classes and Interfaces are true?

|  |  |
| --- | --- |
| Correct Answer | all attributes in an interface, if any, must be constants (i.e. public static final) |
|  | an abstract class doesn’t include implementation (method bodies) for any of its method definitions |
|  | a class can extend more than one abstract class, but a class can implement only one interface |
| Incorrect Response | all of these |
| **Question 10** | |  | 0 / 1 point |

Which of the following is **false**:

|  |  |
| --- | --- |
| Incorrect Response | An interface can extend another Java interface only, an abstract class can extend another Java class and implement multiple Java interfaces. |
|  | Members of a Java interface are public by default. A Java abstract class can have private, protected, etc.. |
|  | A Java interface are implicitly abstract and cannot have implementations. A Java abstract class can have instance methods that implements a default behavior. |
| Correct Answer | Variables declared in a Java abstract class is by default final. An interface may contain non-final variables. |
|  | A Java class can implement multiple interfaces but it can extend only one abstract class. |
| **Question 11** | |  | 0 / 1 point |

Which of the following statements about Abstract Classes and Interfaces are true?

|  |  |
| --- | --- |
|  | all attributes in an abstract class, if any, must be constants (i.e. public static final) |
|  | a class can extend more than one abstract class, but a class can implement only one interface |
| Incorrect Response | all of these |
| Correct Answer | an interface doesn’t include implementation (method bodies) for any of its method definitions |
| **Question 12** | |  | 0 / 1 point |

Which of the following is **false**:

|  |  |
| --- | --- |
|  | Variables declared in a Java interface is by default final. An abstract class may contain non-final variables. |
| Incorrect Response | An interface can extend another Java interface only, an abstract class can extend another Java class and implement multiple Java interfaces. |
|  | A Java class can implement multiple interfaces but it can extend only one abstract class. |
|  | Members of a Java interface are public by default. A Java abstract class can have private, protected, etc.. |
| Correct Answer | A Java abstract class is implicitly abstract and cannot have implementations. A Java interface can have instance methods that implements a default behavior. |
| **Question 13** | |  | 0 / 1 point |

Which of the following is **false**:

|  |  |
| --- | --- |
|  | A Java interface are implicitly abstract and cannot have implementations. A Java abstract class can have instance methods that implements a default behavior. |
|  | Members of a Java interface are public by default. A Java abstract class can have private, protected, etc.. |
| Correct Answer | A Java class can implement one or more interfaces, or extend only one abstract class, but not both |
| Incorrect Response | An interface can extend another Java interface only, an abstract class can extend another Java class and implement multiple Java interfaces. |
|  | Variables declared in a Java interface is by default final. An abstract class may contain non-final variables. |
| **Question 14** | |  | 1 / 1 point |

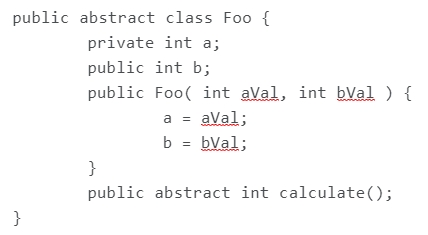
When we are defining a new interface, what happens if we leave out the abstract keyword in the method definitions?

|  |  |
| --- | --- |
|  | A compilation error would result because abstract methods are not allowed in an interface. |
|  | Because the abstract keyword is for abstract classes, a compilation error would result. |
|  | Because all methods in an interface are abstract, it would cause a compilation error. |
|  | Because all methods in an interface are abstract, it doesn’t matter whether or not we specify the abstract keyword. |
| **Question 15** | |  | 1 / 1 point |

Which of the following statements about Abstract Classes and Interfaces are true?

|  |  |
| --- | --- |
|  | a class can extend only one abstract class, but a class can implement more than one interface |
|  | all attributes in an abstract class, if any, must be constants (i.e. public static final) |
|  | an abstract class doesn’t include implementation (method bodies) for any of its method definitions |
|  | all of these |
| **Question 16** | |  | 1 / 1 point |

Consider the abstract superclass below:



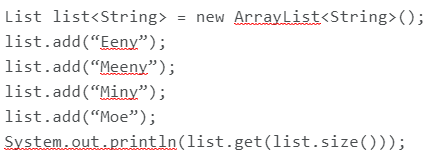
Any *concrete* subclass that *extends* class Foo:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | | a) | Must implement a method called calculate. | |
|  | |  |  | | --- | --- | | b) | Will not be able to access the instance variable a | |
|  | |  |  | | --- | --- | | c) | Neither a) nor b) | |
|  | |  |  | | --- | --- | | d) | Both a) and b) | |
| **Question 17** | |  | 1 / 1 point |

Which of the following is **false**:

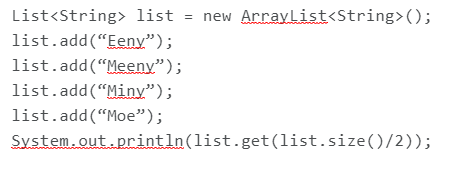
|  |  |
| --- | --- |
|  | A class declared with abstract keyword and have zero or more abstract methods is known as abstract class. |
|  | To use, you need to extend an abstract classes to a concrete class. |
|  | You cannot extend a concrete class to an abstract class |
|  | An abstract class can have any number of constructors |
| **Question 18** | |  | 1 / 1 point |

What is the output of the following code segment:



|  |  |
| --- | --- |
|  | Eeny |
|  | Meeny |
|  | Miny |
|  | Moe |
|  | Throw an exception |
| **Question 19** | |  | 1 / 1 point |

What is the output of the following code segment:



|  |  |
| --- | --- |
|  | Eeny |
|  | Meeny |
|  | Miny |
|  | Moe |
|  | Throw an exception |
| **Question 20** | |  | 0 / 1 point |

Which of the following statements about toString() method of a user-defined class is false:

|  |  |
| --- | --- |
| Correct Answer | can be overloaded if multiple String representations are required |
| Incorrect Response | If you print an object, java compiler internally invokes it |
|  | the default behavior is to return a string containing the class name |
|  | overrides the toString() method of the Object class |
|  | returns the string representation of the object |
| **Question 21** | |  | 1 / 1 point |

Overriding a method differs from overloading a method because:

|  |  |
| --- | --- |
|  | Overloaded methods have the same signature. |
|  | Overridden methods have the same signature. |
|  | Both of the above. |
|  | Neither of the above. |
| **Question 22** | |  | 1 / 1 point |

Which of the following statements is **true**:

|  |  |
| --- | --- |
|  | A static variable cannot be changed |
|  | Polymorphism refers to the bundling of data with the methods that operate on that data |
|  | A List can contain both primitives and reference types |
|  | A multidimensional array can have more than three dimensions |
|  | A package is intended to group related objects |
| **Question 23** | |  | 1 / 1 point |

Which keyword is used to import a package from the Java API library?

|  |  |
| --- | --- |
|  | lib |
|  | getlib |
|  | package |
|  | import |
| **Question 24** | |  | 0 / 1 point |

What is true of class variables in Java that are declared to be static?

|  |  |
| --- | --- |
|  | They are implemented on the stack |
|  | They are immutable and cannot be changed |
| Correct Answer | There is only one copy of a static class variable, shared by all instances |
| Incorrect Response | They are a property of every object that is an instance of the class |
| **Question 25** | |  | 0 / 1 point |

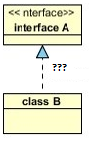
When should a program *explicitly* use the **this** reference?

|  |  |
| --- | --- |
| Incorrect Response | Accessing a local variable |
|  | Accessing a private variable. |
|  | Accessing a public variable |
| Correct Answer | Accessing a field that has the same name as a local variable. |
| **Question 26** | |  | 1 / 1 point |

Which of the following is true about **Objects** in Java?

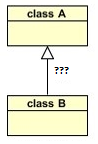
|  |  |
| --- | --- |
|  | Instance variables implement attributes, and constructors implement behaviors. |
|  | Instance variables implement attributes, and methods implement behaviors. |
|  | Local variables implement attributes, and constructors implement behaviors. |
|  | Local variables implement attributes, and methods implement behaviors. |
|  | None of these answers. |
| **Question 27** | |  | 1 / 1 point |

What is the best word for the relationship?



|  |  |
| --- | --- |
|  | contains |
|  | is-a |
|  | owns |
|  | realizes |
|  | uses |
| **Question 28** | |  | 1 / 1 point |

What is the best word for the relationship?



|  |  |
| --- | --- |
|  | realizes |
|  | contains |
|  | owns |
|  | is-a |
|  | uses |
| **Question 29** | |  | 1 / 1 point |

Which symbol is generalization ?

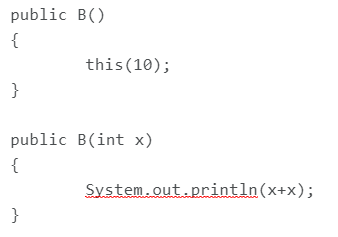
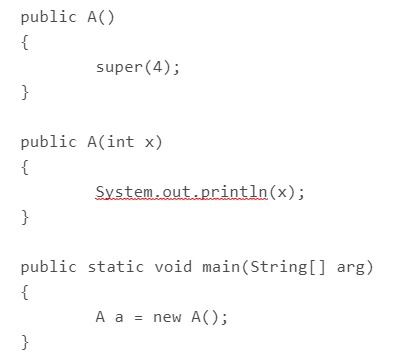
|  |  |
| --- | --- |
|  | Arrow with dotted line |
|  | Filled in diamond |
|  | Empty diamond |
|  | Triagular Arrow |
|  | None-triangular Arrow |
| **Question 30** | |  | 1 / 1 point |

Which symbol is aggregation?

|  |  |
| --- | --- |
|  | None-triangular Arrow |
|  | Filled in diamond |
|  | Empty diamond |
|  | Arrow with dotted line |
|  | Triagular Arrow |
| **Question 31** | |  | 0 / 1 point |

Can a subclass constructor call it's superclass constructor?

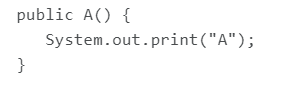
|  |  |
| --- | --- |
| Incorrect Response | Yes, and it can be anywhere in the constructor |
|  | Never |
| Correct Answer | Yes, but the call must be the first statement of the constructor |
|  | Yes, but the call must be the last statement of the constructor |
| **Question 32** | |  | 1 / 1 point |

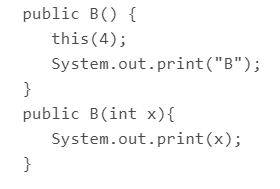
You have a class named B with the following in it:   
  
  
You have a class named A that extends class B, and has the following in it:  


What will be the output?

|  |  |
| --- | --- |
|  | 4 |
|  | 20 |
|  | 8 |
|  | 10 |
|  | Compile error |
| **Question 33** | |  | 1 / 1 point |

You have a class named A with the following in it:



You have a class named B that extends class A, and has the following in it:  


What is printed out by the following Java statement:

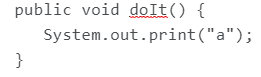
B b = new B();

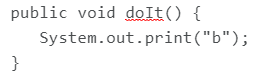
|  |  |
| --- | --- |
|  | AB |
|  | None of these |
|  | A4B |
|  | AB4 |
|  | 4AB |
| **Question 34** | |  | 1 / 1 point |

The default constructor of a subclass always calls the default constructor of it's superclass

|  |  |  |
| --- | --- | --- |
|  |  | True |
|  |  | False |
| **Question 35** | | |  | 1 / 1 point |

You have a class named A with the following in it:



You have a class named B that extends class A, and has the following in it:  


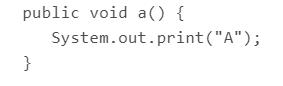
What is printed out by the following Java statements:

A a = new B();

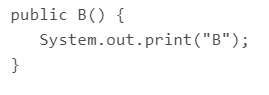
a.doIt();

|  |  |
| --- | --- |
|  | ba |
|  | a |
|  | b |
|  | None of these |
|  | ab |
| **Question 36** | |  | 1 / 1 point |

You have a class named A with the following in it:



You have a class named B that extends class A and has the following in it:

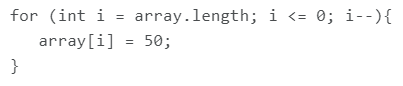


What is printed out by the following Java statement:

A a = new B();

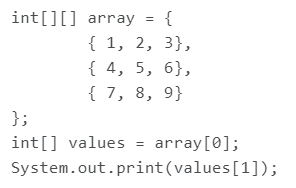
|  |  |
| --- | --- |
|  | B |
|  | BA |
|  | AB |
|  | A |
|  | none of these |
| **Question 37** | |  | 1 / 1 point |

You have an array named array of type int and of size 10. What is true of the following Java code?



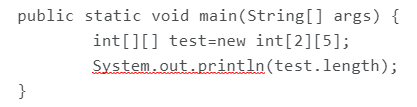
|  |  |
| --- | --- |
|  | it would generate a runtime error |
|  | it would result in a compile-time error |
|  | it would assign all of the elements of array the value 0 |
|  | it would assign all of the elements of array the value 50 |
|  | it would result in an IndexOutOfBoundsException being thrown |
| **Question 38** | |  | 1 / 1 point |

What is result of the following Java code?



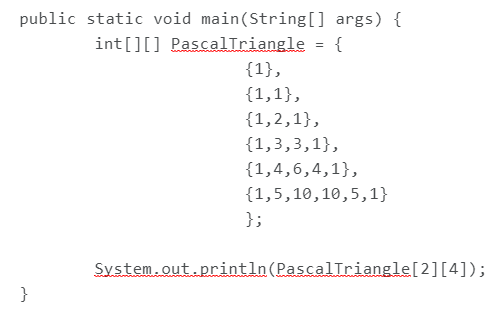
|  |  |
| --- | --- |
|  | 5 |
|  | compiler error |
|  | 2 |
|  | ArrayIndexOutOfBoundsException |
|  | 4 |
| **Question 39** | |  | 1 / 1 point |

What is the output of the following?:



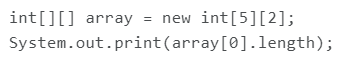
|  |  |
| --- | --- |
|  | runtime error |
|  | 10 |
|  | 2 |
|  | 5 |
|  | nothing, the code won’t compile |
| **Question 40** | |  | 1 / 1 point |

What is the output of the following?:



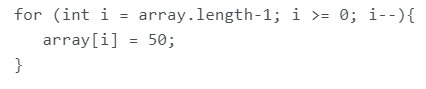
|  |  |
| --- | --- |
|  | nothing, the code won’t compile |
|  | 3 |
|  | 6 |
|  | runtime exception |
|  | 4 |
| **Question 41** | |  | 1 / 1 point |

What is result of the following Java code?



|  |  |
| --- | --- |
|  | 2 |
|  | 5 |
|  | 10 |
|  | 0 |
|  | None of the above |
| **Question 42** | |  | 1 / 1 point |

You have an array named array of type int and of size 10. What is true of the following Java code?



|  |  |
| --- | --- |
|  | it would assign all of the elements of array the value 50 except for the last that will have 0 |
|  | it would assign all of the elements of array the value 0 |
|  | it would generate a runtime error |
|  | it would result in an IndexOutOfBoundsException being thrown |
|  | it would assign all of the elements of array the value 50 |
| **Question 43** | |  | 3 / 4 points |

Match "Subclass Different Package" member to it's visibility

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  | \_\_2\_\_ |  | private | |  | \_\_1\_\_ |  | protected | |  | \_\_1\_\_ |  | public | | Incorrect Response | \_\_1\_\_ | **(2)** | default (no modifier) | |  | |  |  | | --- | --- | | **1**. | Visible | | **2**. | Not visible | | |
| **Question 44** | |  | 4 / 4 points | |

Place the following access levels in order from most restrictive to least restrictive:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | \_\_4\_\_ |  | public |
|  | | \_\_1\_\_ |  | private |
|  | | \_\_2\_\_ |  | default (no modifier) |
|  | | \_\_3\_\_ |  | protected |
|  |  | | | |
|  | | | | |

Quiz Submissions - Midterm 1 - June 24

Haut du formulaire

**Chayma Chadlia Tsouli (username: tsou0009)**

**Attempt 1**

Written: Jun 24, 2021 10:00 AM - Jun 24, 2021 10:59 AM

**Submission View**

Released: Jul 13, 2021 2:56 PM

|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 0 / 1 point |

Which of the following statements about Abstract Classes and Interfaces are true?

|  |  |
| --- | --- |
| Correct Answer | an interface doesn’t include implementation (method bodies) for any of its method definitions |
|  | a class can extend more than one abstract class, but a class can implement only one interface |
| Incorrect Response | all of these |
|  | all attributes in an abstract class, if any, must be constants (i.e. public static final) |
| **Question 2** | |  | 1 / 1 point |

Which of the following is **false**:

|  |  |
| --- | --- |
|  | A Java class can implement one or more interfaces, or extend only one abstract class, but not both |
|  | Members of a Java interface are public by default. A Java abstract class can have private, protected, etc.. |
|  | Variables declared in a Java interface is by default final. An abstract class may contain non-final variables. |
|  | An interface can extend another Java interface only, an abstract class can extend another Java class and implement multiple Java interfaces. |
|  | A Java interface are implicitly abstract and cannot have implementations. A Java abstract class can have instance methods that implements a default behavior. |
| **Question 3** | |  | 0 / 1 point |

When we are defining a new interface, what happens if we leave out the abstract keyword in the method definitions?

|  |  |
| --- | --- |
|  | A compilation error would result because abstract methods are not allowed in an interface. |
|  | Because all methods in an interface are abstract, it would cause a compilation error. |
| Incorrect Response | Because the abstract keyword is for abstract classes, a compilation error would result. |
| Correct Answer | Because all methods in an interface are abstract, it doesn’t matter whether or not we specify the abstract keyword. |
| **Question 4** | |  | 1 / 1 point |

Which of the following is **false**:

|  |  |
| --- | --- |
|  | A Java class can implement multiple interfaces but it can extend only one abstract class. |
|  | An interface can extend another Java interface only, an abstract class can extend another Java class and implement multiple Java interfaces. |
|  | Members of a Java interface are public by default. A Java abstract class can have private, protected, etc.. |
|  | A Java interface are implicitly abstract and cannot have implementations. A Java abstract class can have instance methods that implements a default behavior. |
|  | Variables declared in a Java abstract class is by default final. An interface may contain non-final variables. |
| **Question 5** | |  | 0 / 1 point |

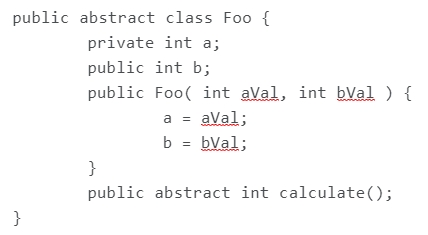
Which of the following statements about Abstract Classes and Interfaces are true?

|  |  |
| --- | --- |
| Correct Answer | all attributes in an interface, if any, must be constants (i.e. public static final) |
| Incorrect Response | all of these |
|  | an abstract class doesn’t include implementation (method bodies) for any of its method definitions |
|  | a class can extend more than one abstract class, but a class can implement only one interface |
| **Question 6** | |  | 1 / 1 point |

Which of the following is **false**:

|  |  |
| --- | --- |
|  | To use you need to extend an abstract class to a concrete class. |
|  | You can extend a concrete class to an abstract class |
|  | An abstract class must have at least one abstract method. |
|  | An abstract class can have any constructors |
| **Question 7** | |  | 0 / 1 point |

Consider the abstract superclass below:



Any *concrete* subclass that *extends* class Foo:

|  |  |  |  |
| --- | --- | --- | --- |
| Incorrect Response | |  |  | | --- | --- | | a) | Must implement a method called calculate. | |
|  | |  |  | | --- | --- | | b) | Will not be able to access the instance variable a | |
|  | |  |  | | --- | --- | | c) | Neither a) nor b) | |
| Correct Answer | |  |  | | --- | --- | | d) | Both a) and b) | |
| **Question 8** | |  | 0 / 1 point |

Which of the following is **false**:

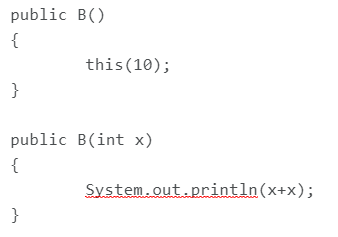
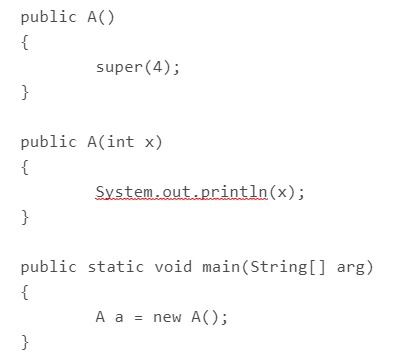
|  |  |
| --- | --- |
| Incorrect Response | A class declared with abstract keyword and have zero or more abstract methods is known as abstract class. |
| Correct Answer | An abstract class can only have a no-arg constructor. |
|  | To use, you need to extend an abstract class to a concrete class. |
|  | You can extend a concrete class to an abstract class |
| **Question 9** | |  | 0 / 1 point |

Which of the following statements is **true:**

|  |  |
| --- | --- |
|  | an interface can have protected instance variables |
|  | an abstract class cannot extend a concrete class |
| Incorrect Response | an abstract class must contain at least one abstract method |
| Correct Answer | an interface can contain constants |
| **Question 10** | |  | 0 / 1 point |

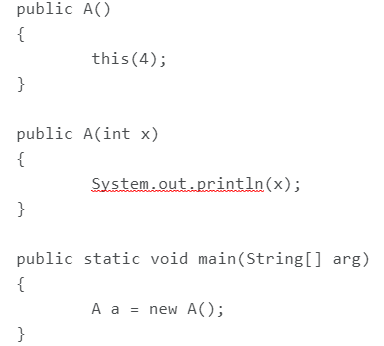
Which of these is the correct way of calling a constructor having no parameters, of superclass A by subclass B?

|  |  |
| --- | --- |
| Incorrect Response | super.A(); |
|  | superclass.(); |
| Correct Answer | super(); |
|  | super(void); |
| **Question 11** | |  | 0 / 1 point |

You have a class named B with the following in it:   
  
  
You have a class named A that extends class B, and has the following in it:  


What will be the output?

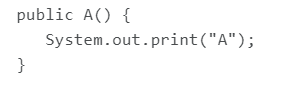
|  |  |
| --- | --- |
|  | 20 |
| Incorrect Response | Compile error |
|  | 4 |
|  | 10 |
| Correct Answer | 8 |
| **Question 12** | |  | 0 / 1 point |

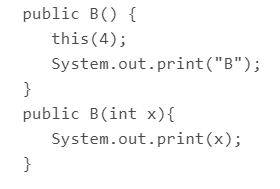
You have a class named A with the following in it:   


What will be the output?

|  |  |
| --- | --- |
| Correct Answer | 4 |
|  | runs with no output |
| Incorrect Response | compile error |
|  | 44 |
|  | runtime error |
| **Question 13** | |  | 1 / 1 point |

You have a class named A with the following in it:



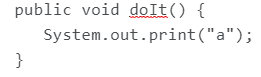
You have a class named B that extends class A, and has the following in it:  


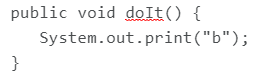
What is printed out by the following Java statement:

B b = new B();

|  |  |
| --- | --- |
|  | A4B |
|  | AB4 |
|  | None of these |
|  | AB |
|  | 4AB |
| **Question 14** | |  | 0 / 1 point |

You have a class named A with the following in it:



You have a class named B that extends class A, and has the following in it:  


What is printed out by the following Java statements:

A a = new B();

a.doIt();

|  |  |
| --- | --- |
| Incorrect Response | None of these |
|  | ab |
|  | ba |
| Correct Answer | b |
|  | a |
| **Question 15** | |  | 1 / 1 point |

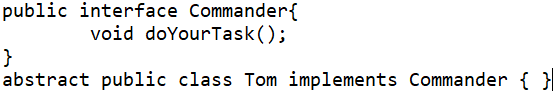
Can a constructor call another constructor of the same class?

|  |  |
| --- | --- |
|  | Yes, and it can be anywhere in the constructor |
|  | Yes, but the call must be the first statement of the constructor |
|  | Yes, but the call must be the last statement of the constructor |
|  | Never |
| **Question 16** | |  | 1 / 1 point |

Choose the class definition that defines a legal abstract class.

|  |  |
| --- | --- |
|  | class D { abstract void incomplete() { }; } |
|  | class D { abstract void incomplete(); } |
|  | abstract class D { abstract void incomplete(); } |
|  | public class abstract D { abstract void incomplete(); } |
| **Question 17** | |  | 1 / 1 point |

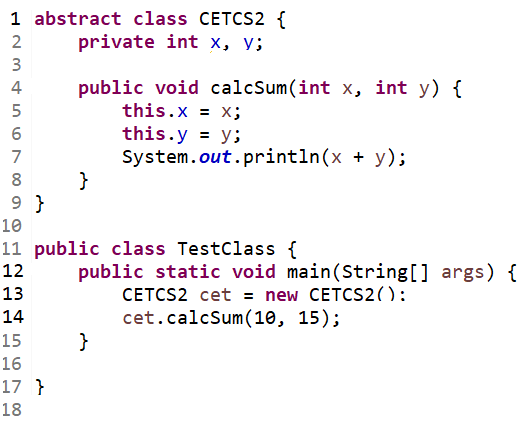
Given the following piece of code:



which of the the statements below is correct?

|  |  |
| --- | --- |
|  | The code won't compile since class Tom must implement method doYourTask() from interface Commander. |
|  | The program will not compile since declaration of class Tom uses keyword implements instead of extends. |
|  | The program will compile without errors. |
|  | The program will not compile since method doYourTask() in interface Commander must be defined as abstract. |
| **Question 18** | |  | 0 / 1 point |

What will happen when this program is compiled?

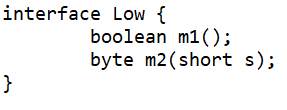


|  |  |
| --- | --- |
|  | None of these |
|  | compilation error due to line 7 |
| Incorrect Response | runtime error due line 2 |
| Correct Answer | compilation error due to line 13 |
|  | run successfully and print 25 |
| **Question 19** | |  | 0 / 1 point |

Declaring a class abstract in Java is useful:

|  |  |
| --- | --- |
|  | When it is meaningful to have objects of that class |
| Incorrect Response | to make programmers to extend that class to use its capabilities |
|  | when it is not desirable to have default implementations of some of the methods |
| Correct Answer | In situations where it is meaningless to create objects of that class. |
|  | To keep programmers from extending the class. |
| **Question 20** | |  | 0 / 1 point |

Based on this code:

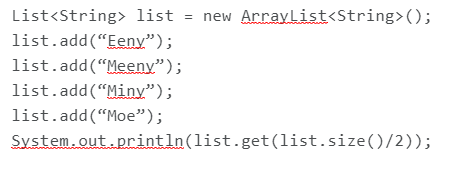


which of the two code fragments shown below will compile?

1. interface High implements Low{}  
2. abstract class MidPoint extends Low  
    { public boolean m1(){ return true;}  
3. abstract class MidPoint implements Low{}  
4. abstract class MidPoint implements Low  
    {public boolean m1(){ return (8 > 5);}  
5. abstract class MidPoint implements Low  
    {protected boolean m1() { return (3 < 5);}

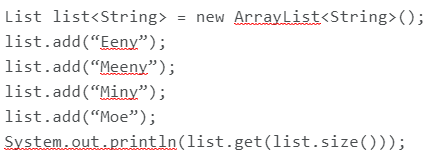
|  |  |
| --- | --- |
|  | 1 and 2 |
|  | 2 and 3 |
| Correct Answer | 3 and 4 |
| Incorrect Response | 1 and 3 |
|  | 4 and 5 |
| **Question 21** | |  | 1 / 1 point |

What is the output of the following code segment:



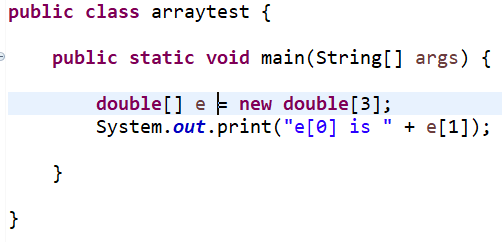
|  |  |
| --- | --- |
|  | Eeny |
|  | Meeny |
|  | Miny |
|  | Moe |
|  | Throw an exception |
| **Question 22** | |  | 1 / 1 point |

What is the output of the following code segment:



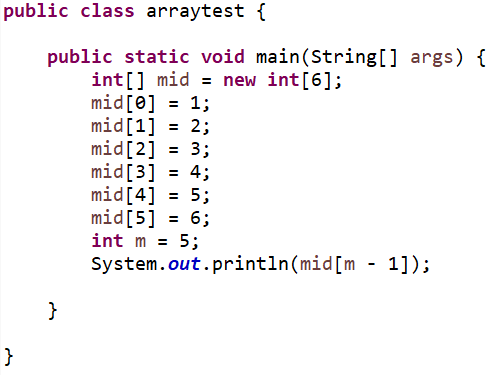
|  |  |
| --- | --- |
|  | Eeny |
|  | Meeny |
|  | Miny |
|  | Moe |
|  | Throw an exception |
| **Question 23** | |  | 1 / 1 point |

What is does the program below output?



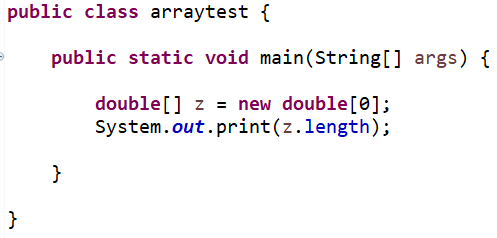
|  |  |
| --- | --- |
|  | a compile time error since an incorrect array size was specified in the array declaration |
|  | The program compiles and runs fine and displaying e[0] is 0.0 as output |
|  | a runtime error caused by e [1] in the output statement |
|  | a runtime error caused by failing to initialize array elements |
| **Question 24** | |  | 1 / 1 point |

What is the output of the code fragment shown below?



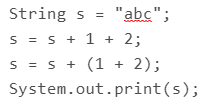
|  |  |
| --- | --- |
|  | 3 |
|  | 4 |
|  | 2 |
|  | 5 |
| **Question 25** | |  | 1 / 1 point |

What is the output of the following program code?



|  |  |
| --- | --- |
|  | none of the above and or below |
|  | Compile-time error, should use z.length() instead of z.length |
|  | compile-time error, arrays cannot be initialized to zero |
|  | 0 |
| **Question 26** | |  | 0 / 1 point |

What is the output of the following code:



|  |  |
| --- | --- |
| Correct Answer | abc123 |
|  | abc33 |
|  | abc312 |
| Incorrect Response | abc1212 |
| **Question 27** | |  | 1 / 1 point |

If a parent class has an abstract method called “MyMethod” then the concrete child classes:

|  |  |
| --- | --- |
|  | May implement MyMethod and the parameters may or may not match |
|  | May implement MyMethod but only if the parameters match |
|  | May implement MyMethod but only if the parameters do not match |
|  | Must not implement MyMethod |
|  | Must implement MyMethod |
| **Question 28** | |  | 1 / 1 point |

Which keyword is used to import a package from the Java API library?

|  |  |
| --- | --- |
|  | import |
|  | package |
|  | getlib |
|  | lib |
| **Question 29** | |  | 0 / 1 point |

Which of the following statements is **true**:

|  |  |
| --- | --- |
| Incorrect Response | Polymorphism refers to the bundling of data with the methods that operate on that data |
|  | A List can contain both primitives and reference types |
|  | A static variable cannot be changed |
| Correct Answer | A package is intended to group related classes |
|  | A multidimensional array cannot have more than three dimensions |
| **Question 30** | |  | 0 / 1 point |

If a parent class has a public method called “MyMethod” then the child classes:

|  |  |
| --- | --- |
| Correct Answer | May implement MyMethod and the parameters may or may not match |
|  | May implement MyMethod but only if the parameters match |
|  | May implement MyMethod but only if the parameters do not match |
| Incorrect Response | Must not implement MyMethod |
|  | Must implement MyMethod |
| **Question 31** | |  | 1 / 1 point |

Which of the following statements is **true**:

|  |  |
| --- | --- |
|  | Polymorphism refers to the bundling of data with the methods that operate on that data |
|  | A final variable cannot be changed |
|  | A package is intended to group related objects |
|  | A multidimensional array cannot have more than three dimensions |
|  | A List can contain both primitives and reference types |
| **Question 32** | |  | 1 / 1 point |

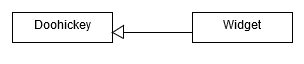
Which method can be used to find the length of a string?

|  |  |
| --- | --- |
|  | getLength() |
|  | getSize() |
|  | length() |
|  | len() |
|  | size() |
| **Question 33** | |  | 1 / 1 point |

The difference between aggregation and composition is:

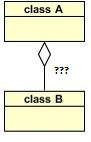
|  |  |
| --- | --- |
|  | There are the same thing |
|  | Aggregation is a stronger form of composition |
|  | Composition is a stronger form of aggregation |
|  | They are not related |
| **Question 34** | |  | 0 / 1 point |

What does the following mean?



|  |  |
| --- | --- |
|  | A Widget is a Doohickey |
|  | A Widget has a Doohickey |
|  | A Widget use a Doohickey |
|  | A Doohickey is a Widget |
|  | A Doohickey has a Widget |
| **Question 35** | |  | 0 / 1 point |

What is the best word for the relationship?

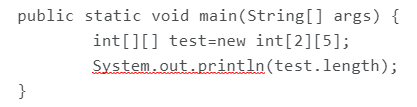


|  |  |
| --- | --- |
|  | uses |
|  | realizes |
|  | is-a |
|  | contains |
|  | owns |
| **Question 36** | |  | 0 / 1 point |

Which symbol is association?

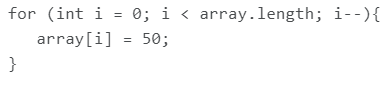
|  |  |
| --- | --- |
|  | Filled in diamond |
|  | Empty diamond |
|  | None-triangular Arrow |
|  | Triagular Arrow |
|  | Arrow with dotted line |
| **Question 37** | |  | 0 / 1 point |

What is the output of the following?:



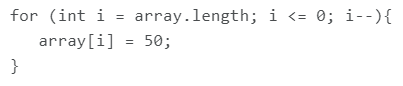
|  |  |
| --- | --- |
|  | 10 |
|  | 5 |
|  | 2 |
|  | runtime error |
|  | nothing, the code won’t compile |
| **Question 38** | |  | 0 / 1 point |

You have an array named array of type int and of size 10. What is true of the following Java code?



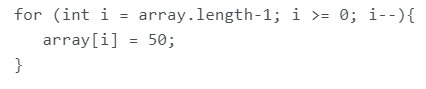
|  |  |
| --- | --- |
|  | it would assign all of the elements of array the value 50 |
|  | it would assign all of the elements of array the value 0 |
|  | it would result in an IndexOutOfBoundsException being thrown |
|  | it would result in a compile-time error |
| **Question 39** | |  | 0 / 1 point |

You have an array named array of type int and of size 10. What is true of the following Java code?



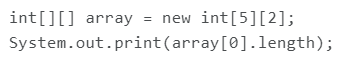
|  |  |
| --- | --- |
|  | it would generate a runtime error |
|  | it would result in a compile-time error |
|  | it would assign all of the elements of array the value 0 |
|  | it would result in an IndexOutOfBoundsException being thrown |
|  | it would assign all of the elements of array the value 50 |
| **Question 40** | |  | 0 / 1 point |

You have an array named array of type int and of size 10. What is true of the following Java code?



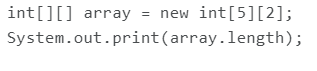
|  |  |
| --- | --- |
|  | it would result in an IndexOutOfBoundsException being thrown |
|  | it would assign all of the elements of array the value 0 |
|  | it would assign all of the elements of array the value 50 except for the last that will have 0 |
|  | it would generate a runtime error |
|  | it would assign all of the elements of array the value 50 |
| **Question 41** | |  | 0 / 1 point |

What is result of the following Java code?



|  |  |
| --- | --- |
|  | 2 |
|  | 5 |
|  | 10 |
|  | 0 |
|  | None of the above |
| **Question 42** | |  | 0 / 1 point |

What is result of the following Java code?



|  |  |
| --- | --- |
|  | 2 |
|  | 5 |
|  | 10 |
|  | 0 |
|  | None of the above |
| **Question 43** | |  | 0 / 4 points |

Match "Subclass Diff Pkg." member to it's visibility

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | Incorrect Response | \_\_\_\_\_ | **(2)** | private | | Incorrect Response | \_\_\_\_\_ | **(1)** | public | | Incorrect Response | \_\_\_\_\_ | **(1)** | protected | | Incorrect Response | \_\_\_\_\_ | **(2)** | default (no modifier) | |  | |  |  | | --- | --- | | **1**. | Visible | | **2**. | Not visible | | |
| **Question 44** | |  | 0 / 4 points | |

Match "Subclass Same Pkg." members to it's visibility

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | Incorrect Response | \_\_\_\_\_ | **(1)** | public | | Incorrect Response | \_\_\_\_\_ | **(1)** | protected | | Incorrect Response | \_\_\_\_\_ | **(2)** | private | | Incorrect Response | \_\_\_\_\_ | **(1)** | default (no modifier) | |  | |  |  | | --- | --- | | **1**. | Visible | | **2**. | Not visible | |
|  |  | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 1 / 1 point |

What would the following program code output?

Text

Description automatically generated

|  |  |
| --- | --- |
|  | 1 |
|  | 2 |
|  | 3 |
|  | 6 |
| **Question 2** | |  | 1 / 1 point |

What is the output of the code fragment shown below?

Text

Description automatically generated with medium confidence

|  |  |
| --- | --- |
|  | 3 |
|  | 4 |
|  | 5 |
|  | 2 |
| **Question 3** | |  | 1 / 1 point |

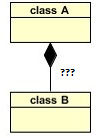
What is does the program below output?

Text

Description automatically generated

|  |  |
| --- | --- |
|  | a runtime error caused by e [1] in the output statement |
|  | The program compiles and runs fine and displaying e[0] is 0.0 as output |
|  | a runtime error caused by failing to initialize array elements |
|  | a compile time error since an incorrect array size was specified in the array declaration |
| **Question 4** | |  | 0 / 1 point |

What is the best word for the relationship?



|  |  |
| --- | --- |
|  | realizes |
|  | contains |
| Correct Answer | owns |
| Incorrect Response | is-a |
|  | uses |
| **Question 5** | |  | 1 / 1 point |

Which symbol is realization ?

|  |  |
| --- | --- |
|  | Empty diamond |
|  | Arrow with dotted line |
|  | None-triangular Arrow |
|  | Triagular Arrow |
|  | Filled in diamond |
| **Question 6** | |  | 1 / 1 point |

The statement “class DoHickey extends ThingaBob” means?

|  |  |
| --- | --- |
|  | ThingaBob inherits from DoHickey |
|  | all of the others |
|  | ThingaBob is-a DoHickey |
|  | ThingaBob is a generalization of DoHickey |
| **Question 7** | |  | 1 / 1 point |

Which symbol is generalization ?

|  |  |
| --- | --- |
|  | Filled in diamond |
|  | Arrow with dotted line |
|  | Empty diamond |
|  | Triagular Arrow |
|  | None-triangular Arrow |
| **Question 8** | |  | 1 / 1 point |

You have a class named A with the following in it:

Text

Description automatically generated with low confidence

You have a class named B that extends class A, and has the following in it:  
Text

Description automatically generated

What is printed out by the following Java statement:

A a = new B();

|  |  |
| --- | --- |
|  | 1B |
|  | 1 |
|  | None of these |
|  | B |
|  | xB |
| **Question 9** | |  | 1 / 1 point |

You have a class named A with the following in it:

A picture containing logo

Description automatically generated

You have a class named B that extends class A, and has the following in it:  
A picture containing company name

Description automatically generated

What is printed out by the following Java statements:

A a = new B();

a.doIt();

|  |  |
| --- | --- |
|  | ab |
|  | a |
|  | b |
|  | None of these |
|  | ba |
| **Question 10** | |  | 1 / 1 point |

Which of these is the correct way of calling a constructor having no parameters, of superclass A by subclass B?

|  |  |
| --- | --- |
|  | super(); |
|  | super(void); |
|  | superclass.(); |
|  | super.A(); |
| **Question 11** | |  | 1 / 1 point |

You have a class named A with the following in it:

Text

Description automatically generated with low confidence

You have a class named B that extends class A and has the following in it:

A picture containing text

Description automatically generated

What is printed out by the following Java statement:

A a = new B();

|  |  |
| --- | --- |
|  | AB |
|  | BA |
|  | none of these |
|  | A |
|  | B |
| **Question 12** | |  | 0 / 1 point |

Which of these statements is **false**:

|  |  |
| --- | --- |
|  | this() and super() must be the first statement in the constructor |
|  | We cannot use this() and super() in the same constructor |
| Correct Answer | None of these |
| Incorrect Response | this() can call same class constructor only |
|  | super() can call immediate super class constructor only |
| **Question 13** | |  | 1 / 1 point |

Can a constructor call another constructor of the same class?

|  |  |
| --- | --- |
|  | Yes, and it can be anywhere in the constructor |
|  | Never |
|  | Yes, but the call must be the last statement of the constructor |
|  | Yes, but the call must be the first statement of the constructor |
| **Question 14** | |  | 4 / 4 points |

Match "Subclass Same Pkg." members to it's visibility

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  | \_\_1\_\_ |  | protected | |  | \_\_2\_\_ |  | private | |  | \_\_1\_\_ |  | public | |  | \_\_1\_\_ |  | default (no modifier) | |  | |  |  | | --- | --- | | **1**. | Visible | | **2**. | Not visible | | |
| **Question 15** | |  | 4 / 4 points | |

Match "Subclass Diff Pkg." member to it's visibility

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  | \_\_1\_\_ |  | protected | |  | \_\_2\_\_ |  | default (no modifier) | |  | \_\_1\_\_ |  | public | |  | \_\_2\_\_ |  | private | |  | |  |  | | --- | --- | | **1**. | Visible | | **2**. | Not visible | | |
| **Question 16** | |  | 1 / 1 point | |

Which of the following statements is **true:**

|  |  |
| --- | --- |
|  | an abstract class cannot extend a concrete class |
|  | an interface can contain constants |
|  | an interface can have protected instance variables |
|  | an abstract class must contain at least one abstract method |
| **Question 17** | |  | 1 / 1 point |

Which of the following is **false**:

|  |  |
| --- | --- |
|  | A Java class can implement multiple interfaces but it can extend only one abstract class. |
|  | A Java interface are implicitly abstract and cannot have implementations. A Java abstract class can have instance methods that implements a default behavior. |
|  | Variables declared in a Java interface is by default final. An abstract class may contain non-final variables. |
|  | Members of an abstract class are public by default. An interface can have private, protected, etc.. |
|  | An interface can extend another Java interface only, an abstract class can extend another Java class and implement multiple Java interfaces. |
| **Question 18** | |  | 1 / 1 point |

Which of the following statements about Abstract Classes and Interfaces are true?

|  |  |
| --- | --- |
|  | all of these |
|  | all attributes in an abstract class, if any, must be constants (i.e. public static final) |
|  | a class can extend only one abstract class, but a class can implement more than one interface |
|  | an abstract class doesn’t include implementation (method bodies) for any of its method definitions |
| **Question 19** | |  | 1 / 1 point |

Which of the following statements about Abstract Classes and Interfaces are true?

|  |  |
| --- | --- |
|  | all attributes in an interface, if any, must be constants (i.e. public static final) |
|  | a class can extend only one abstract class, but a class can implement more than one interface |
|  | all of the these |
|  | an interface doesn’t include implementation (method bodies) for any of its method definitions |
| **Question 20** | |  | 1 / 1 point |

Which of the following statements about Abstract Classes and Interfaces are true?

|  |  |
| --- | --- |
|  | all of these |
|  | all attributes in an abstract class, if any, must be constants (i.e. public static final) |
|  | a class can extend more than one abstract class, but a class can implement only one interface |
|  | an interface doesn’t include implementation (method bodies) for any of its method definitions |
| **Question 21** | |  | 1 / 1 point |

Consider the abstract superclass below:

Graphical user interface, text, application

Description automatically generated

Any *concrete* subclass that *extends* class Foo:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | | a) | Must implement a method called calculate. | |
|  | |  |  | | --- | --- | | b) | Will not be able to access the instance variable a | |
|  | |  |  | | --- | --- | | c) | Neither a) nor b) | |
|  | |  |  | | --- | --- | | d) | Both a) and b) | |
| **Question 22** | |  | 1 / 1 point |

Which of the following statements about Abstract Classes and Interfaces are true?

|  |  |
| --- | --- |
|  | an abstract class doesn’t include implementation (method bodies) for any of its method definitions |
|  | a class can extend more than one abstract class, but a class can implement only one interface |
|  | all attributes in an interface, if any, must be constants (i.e. public static final) |
|  | all of these |
| **Question 23** | |  | 1 / 1 point |

What Java keyword is used to inherit an interface?

|  |  |
| --- | --- |
|  | implements |
|  | overrides |
|  | overloads |
|  | inherits |
|  | extends |
| **Question 24** | |  | 1 / 1 point |

Which of the following is **false**:

|  |  |
| --- | --- |
|  | A class declared with abstract keyword and have zero or more abstract methods is known as abstract class. |
|  | To use, you need to extend an abstract class to a concrete class. |
|  | An abstract class can only have a no-arg constructor. |
|  | You can extend a concrete class to an abstract class |
| **Question 25** | |  | 1 / 1 point |

What is result of the following Java code?

Text

Description automatically generated with medium confidence

|  |  |
| --- | --- |
|  | 2 |
|  | 5 |
|  | 10 |
|  | 0 |
|  | None of the above |
| **Question 26** | |  | 1 / 1 point |

You have an array named array of type int and of size 10. What is true of the following Java code?

A picture containing text

Description automatically generated

|  |  |
| --- | --- |
|  | it would generate a runtime error |
|  | it would assign all of the elements of array the value 50 except for the last that will have 0 |
|  | it would result in an IndexOutOfBoundsException being thrown |
|  | it would assign all of the elements of array the value 50 |
|  | it would assign all of the elements of array the value 0 |
| **Question 27** | |  | 1 / 1 point |

You have an array named array of type int and of size 10. What is true of the following Java code?

A picture containing calendar

Description automatically generated

|  |  |
| --- | --- |
|  | it would assign all of the elements of array the value 0 |
|  | it would assign all of the elements of array the value 50 |
|  | it would result in an IndexOutOfBoundsException being thrown |
|  | it would result in a compile-time error |
| **Question 28** | |  | 0 / 1 point |

What is the output of the following?:

Text

Description automatically generated

|  |  |
| --- | --- |
| Correct Answer | 6 |
|  | runtime exception |
| Incorrect Response | 3 |
|  | 4 |
|  | nothing, the code won’t compile |
| **Question 29** | |  | 1 / 1 point |

What is result of the following Java code?

Text

Description automatically generated with medium confidence

|  |  |
| --- | --- |
|  | 5 |
|  | compiler error |
|  | 4 |
|  | ArrayIndexOutOfBoundsException |
|  | 2 |
| **Question 30** | |  | 1 / 1 point |

You have an array named array of type int and of size 10. What is true of the following Java code?

A picture containing text

Description automatically generated

|  |  |
| --- | --- |
|  | it would result in an IndexOutOfBoundsException being thrown |
|  | it would generate a runtime error |
|  | it would assign all of the elements of array the value 0 |
|  | it would assign all of the elements of array the value 50 |
|  | it would result in a compile-time error |
| **Question 31** | |  | 1 / 1 point |

Which of the following statements is **true**:

|  |  |
| --- | --- |
|  | A multidimensional array cannot have more than three dimensions |
|  | Encapsulation refers to the bundling of data with the methods that operate on that data |
|  | A List can contain both primitives and reference types |
|  | A package is intended to group related objects |
|  | A static variable cannot be changed |
| **Question 32** | |  | 1 / 1 point |

When should a program *explicitly* use the **this** reference?

|  |  |
| --- | --- |
|  | Accessing a private variable. |
|  | Accessing a field that has the same name as a local variable. |
|  | Accessing a local variable |
|  | Accessing a public variable |
| **Question 33** | |  | 1 / 1 point |

When working with a Scanner in Java, its next method will return

|  |  |
| --- | --- |
|  | None of these answers |
|  | The next integer value |
|  | The next word of input |
|  | The next line of input |
|  | The next Unicode character |
| **Question 34** | |  | 1 / 1 point |

Which method can be used to find the length of a string?

|  |  |
| --- | --- |
|  | getLength() |
|  | size() |
|  | getSize() |
|  | len() |
|  | length() |
| **Question 35** | |  | 1 / 1 point |

Which of the following statements about toString() method of a user-defined class is false:

|  |  |
| --- | --- |
|  | the default behavior is to return a string containing the class name |
|  | overrides the toString() method of the Object class |
|  | can be overloaded if multiple String representations are required |
|  | returns the string representation of the object |
|  | If you print an object, java compiler internally invokes it |
| **Question 36** | |  | 1 / 1 point |

Overriding a method differs from overloading a method because:

|  |  |
| --- | --- |
|  | Overloaded methods have the same signature. |
|  | Overridden methods have the same signature. |
|  | Both of the above. |
|  | Neither of the above. |
| **Question 37** | |  | 1 / 1 point |

What is the output of the following code:

A picture containing calendar

Description automatically generated

|  |  |
| --- | --- |
|  | abc312 |
|  | abc123 |
|  | abc1212 |
|  | abc33 |
| **Question 38** | |  | 1 / 1 point |

Which of these is a correct interface?

|  |  |
| --- | --- |
|  | interface B { void print () { } } |
|  | abstract interface B { print (); } |
|  | interface B { void print (); } |
|  | abstract interface B { abstract void print (); { } } |
| **Question 39** | |  | 1 / 1 point |

What is the output of the program below?

Text

Description automatically generated

|  |  |
| --- | --- |
|  | none of these |
|  | Class Learner1 doIt\_1 |
|  | gets compiled fully but prints nothing |
|  | Compilation error |
|  | Class Learner2 doIt\_2 |
| **Question 40** | |  | 1 / 1 point |

Declaring a class abstract in Java is useful:

|  |  |
| --- | --- |
|  | In situations where it is meaningless to create objects of that class. |
|  | to make programmers to extend that class to use its capabilities |
|  | To keep programmers from extending the class. |
|  | when it is not desirable to have default implementations of some of the methods |
|  | When it is meaningful to have objects of that class |
| **Question 41** | |  | 1 / 1 point |

Given this block of code

Text

Description automatically generated

Which of the assertions below is true?

|  |  |
| --- | --- |
|  | You must add a return statement in method numOfKids(). |
|  | keywords abstract and public cannot be used together. |
|  | Class kindergarten must be defined as abstract. |
|  | The method numOfKids() in class kindergarten must have a body. |
| **Question 42** | |  | 0 / 1 point |

Based on this code:

Text

Description automatically generated

which of the two code fragments shown below will compile?

1. interface High implements Low{}  
2. abstract class MidPoint extends Low  
    { public boolean m1(){ return true;}  
3. abstract class MidPoint implements Low{}  
4. abstract class MidPoint implements Low  
    {public boolean m1(){ return (8 > 5);}  
5. abstract class MidPoint implements Low  
    {protected boolean m1() { return (3 < 5);}

|  |  |
| --- | --- |
|  | 2 and 3 |
| Incorrect Response | 4 and 5 |
|  | 1 and 3 |
| Correct Answer | 3 and 4 |
|  | 1 and 2 |
| **Question 43** | |  | 1 / 1 point |

What would be printed by executing the following Java code (assume that all required import statements are included)?

Text

Description automatically generated with medium confidence

|  |  |
| --- | --- |
|  | 0 |
|  | 3 |
|  | Nothing, the code will not compile |
|  | None of these answers |
|  | -1 |
| **Question 44** | |  | 1 / 1 point |

What is the output of the following code segment:

Graphical user interface, text, application

Description automatically generated

|  |  |  |
| --- | --- | --- |
|  | Eeny | |
|  | Meeny | |
|  | Miny | |
|  | Moe | |
|  | Throw an exception | |
|  |  | |
|  | | |
| **Question 1** |  | 1 / 1 point |

Interface List is implemented by the following classes

|  |  |  |  |
| --- | --- | --- | --- |
|  | ArrayList, LinkedList and  Deque | | |
|  | ArrayList, LinkedList and  Vector | | |
|  | ArrayList, LinkedList and  HashSet | | |
|  | ArrayList, LinkedList and  TreeSet | | |
| **Question 2** | |  | 1 / 1 point | |

Collections can manipulate variables of primitive types

|  |  |  |  |
| --- | --- | --- | --- |
|  | true | | |
|  | false | | |
| **Question 3** | |  | 0 / 1 point | |

A set is a collection that can contain duplicates

|  |  |  |  |
| --- | --- | --- | --- |
|  | true | | |
|  | false | | |
| **Question 4** | |  | 0 / 1 point | |

When the ArrayList grows, it must create a larger internal array and copy each  element to the new array.

|  |  |  |  |
| --- | --- | --- | --- |
|  | true | | |
|  | false | | |
| **Question 5** | |  | 0 / 1 point | |

Converting an object of a type-wrapper class to a value of the  corresponding primitive type is called

|  |  |  |  |
| --- | --- | --- | --- |
|  | None of these | | |
|  | type conversion | | |
|  | boxing conversion | | |
|  | unboxing conversion | | |
| **Question 6** | |  | 1 / 1 point | |

A list is an ordered collection that does not contain duplicates

|  |  |  |  |
| --- | --- | --- | --- |
|  | true | | |
|  | false | | |
| **Question 7** | |  | 1 / 1 point | |

Interface set, queue and list are derived from

|  |  |  |  |
| --- | --- | --- | --- |
|  | HashSet interface | | |
|  | ArrayList interface | | |
|  | Dequeue interface | | |
|  | Collection interface | | |
| **Question 8** | |  | 1 / 1 point | |

Interface Collection provides a method that returns an Iterator object, which allows a program to  walk through the collection and remove elements from the collection during the iteration.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | false |
|  | | | true |
|  |  |
|  | |

|  |  |  |
| --- | --- | --- |
| **Question 1**Correct on previous attempt(s) |  | 1 / 1 point |

In what kind of memory are Java local variables stored?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | objects in heap, primitives on stack | | | |
|  | |  |  | | --- | --- | |  | objects on stack, primitives in heap | | | |
|  | |  |  | | --- | --- | |  | stack | | | |
|  | |  |  | | --- | --- | |  | heap | | | |
| **Question 2**Correct on previous attempt(s) | |  | 1 / 1 point | |

In the following Java code fragment:

**try {**

**System.out.print("A");**

**} catch (Exception ex){**

**System.out.print("B");**

**} finally {**

**System.out.print("C");**

**}**

what is printed out assuming no exception occurs?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **AB** | | | |
|  | |  |  | | --- | --- | |  | **AC** | | | |
|  | |  |  | | --- | --- | |  | **A** | | | |
|  | |  |  | | --- | --- | |  | **C** | | | |
|  | |  |  | | --- | --- | |  | **B** | | | |
| **Question 3**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given this class definition

**public class A {**

**public void doit() {**

**System.out.print("done");**

**throw new Exception("ex");**

**}**

**}**

what would be printed by the following Java statements?

**A a = new A();**

**a.doit();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **ex** | | | |
|  | |  |  | | --- | --- | |  | **done** | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile. | | | |
|  | |  |  | | --- | --- | |  | **doneex** | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
| **Question 4**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions:

**public class A {**

**public A(int x) {**

**System.out.print("A");**

**}**

**}**

**public class B extends A {**

**public B() {**

**System.out.print("B");**

**}**

**}**

what is printed out by the following Java statement:

**A a = new B();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **A** | | | |
|  | |  |  | | --- | --- | |  | this code will not compile | | | |
|  | |  |  | | --- | --- | |  | **BA** | | | |
|  | |  |  | | --- | --- | |  | **AB** | | | |
|  | |  |  | | --- | --- | |  | **B** | | | |
| **Question 5**Correct on previous attempt(s) | |  | 1 / 1 point | |

In the following Java code fragment:

**public class MyClass {**

**int x;**

**public void a() {**

**int x;**

**x = 17;**

**}**

**public void b() {**

**a();**

**/\* state of x here?\*/**

**}**

**}**

what is the best description of the state of the variable **x** at the indicated point?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **x** will have no value | | | |
|  | |  |  | | --- | --- | |  | **x** will be available to be garbage collected | | | |
|  | |  |  | | --- | --- | |  | **x** will have value **17** | | | |
|  | |  |  | | --- | --- | |  | **x** will have value **0** | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
| **Question 6**Retaken | |  | 1 / 1 point | |

Which of the following code fragments includes a declaration that breaks the Google Java conventions?

Fragment One:

**int level; // indentation level**

**int size; // size of table**

Fragment Two:

**int   level;      // indentation level**

**int   size;      // size of table**

**Object currentEntry;  // currently selected table entry**

Fragment Three:

**void myMethod() {**

**int int1 = 0; // beginning of method block**

**if (condition) {**

**int int2 = 4, int3 = 0; // beginning of block**

**...**

**}**

**}**

Fragment Four:

**myMethod() {**

**if (condition) {**

**int count = 0;**

**...**

**}**

**...**

**}**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | Fragment Two | | | |
|  | |  |  | | --- | --- | |  | Fragment Three | | | |
|  | |  |  | | --- | --- | |  | Fragment Four | | | |
|  | |  |  | | --- | --- | |  | Fragment One | | | |
| **Question 7**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions:

**public class A {**

**public void doit() {**

**System.out.print("a");**

**}**

**}**

**public class B extends A {**

**public void doit() {**

**System.out.print("b");**

**}**

**}**

what is printed out by the following Java statements:

**A b = new B();**

**b.doit();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **a** | | | |
|  | |  |  | | --- | --- | |  | **ba** | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
|  | |  |  | | --- | --- | |  | **b** | | | |
|  | |  |  | | --- | --- | |  | **ab** | | | |
| **Question 8**Correct on previous attempt(s) | |  | 1 / 1 point | |

Java Interfaces are useful because

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | they can help define an Application Programming Interface (API) | | | |
|  | |  |  | | --- | --- | |  | all of these answers | | | |
|  | |  |  | | --- | --- | |  | they can help coordinate between two separate groups of programmers when one group needs to use methods that the other group implements | | | |
|  | |  |  | | --- | --- | |  | classes can implement more than one interface, which is like a form of multiple inheritance | | | |
|  | |  |  | | --- | --- | |  | an interface forms a contract specifiying what behavior is provided by a class that implements the interface without forcing that behavior to be implemented in a certain way | | | |
| **Question 9**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given the following anonymous class object:

**new Bar(){**

**<java code>**

**}**

assuming it and the **<java code>** are correct, what do we know about **Bar**?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **Bar** is a method name. | | | |
|  | |  |  | | --- | --- | |  | **Bar** is an interface name. | | | |
|  | |  |  | | --- | --- | |  | None of these answers. | | | |
|  | |  |  | | --- | --- | |  | **Bar** is an interface name or a class name. | | | |
|  | |  |  | | --- | --- | |  | **Bar** is a class name. | | | |
| **Question 10**Correct on previous attempt(s) | |  | 1 / 1 point | |

What is true of a class that is declared to be **final**?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **final** classes cannot be instantiated | | | |
|  | |  |  | | --- | --- | |  | **final** classes cannot be extended | | | |
|  | |  |  | | --- | --- | |  | **final** classes have been deprecated, so an alternative should be used instead | | | |
|  | |  |  | | --- | --- | |  | **final** classes cannot extend another class | | | |
| **Question 11**Correct on previous attempt(s) | |  | 1 / 1 point | |

Which of the following would put the elements of a **List<String> myLst** into random order?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **Collections.shuffle(myLst);** | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | **myLst.shuffle();** | | | |
|  | |  |  | | --- | --- | |  | **Collections.random(myLst);** | | | |
|  | |  |  | | --- | --- | |  | **myLst.random();** | | | |
| **Question 12**Correct on previous attempt(s) | |  | 1 / 1 point | |

What is true of the following Java code?

**int[] array = new int[10];**

**for (int i = array.length - 1; i >= 0; i--){**

**array[i] = 50;**

**}**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | it would result in a compile time error | | | |
|  | |  |  | | --- | --- | |  | it would result in an array with ten elements of value **0** | | | |
|  | |  |  | | --- | --- | |  | it would assign all of the elements of **array** the value **50** | | | |
|  | |  |  | | --- | --- | |  | it would generate a runtime error | | | |
|  | |  |  | | --- | --- | |  | it would result in an infinite loop | | | |
| **Question 13**Correct on previous attempt(s) | |  | 1 / 1 point | |

In the following Java code fragment:

**try {**

**int i;**

**System.out.print("A");**

**i = 13 / 0;**

**System.out.print("B");**

**} catch (Exception e){**

**System.out.print("C");**

**} finally {**

**System.out.print("D");**

**}**

what is printed out, assuming the division by zero generates an exception?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **ACD** | | | |
|  | |  |  | | --- | --- | |  | **AC** | | | |
|  | |  |  | | --- | --- | |  | **ADCB** | | | |
|  | |  |  | | --- | --- | |  | **AD** | | | |
|  | |  |  | | --- | --- | |  | **CD** | | | |
| **Question 14**Correct on previous attempt(s) | |  | 1 / 1 point | |

What is true of class variables in Java that are declared to be static?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | they are implemented on the stack | | | |
|  | |  |  | | --- | --- | |  | they are immutable and cannot be changed | | | |
|  | |  |  | | --- | --- | |  | they are a property of every object that is an instance of the class | | | |
|  | |  |  | | --- | --- | |  | they are associated with the class in which they are declared as opposed to objects that are instances of the class | | | |
| **Question 15**Correct on previous attempt(s) | |  | 1 / 1 point | |

What is true of the following Java code?

**int[] array = new int[10];**

**for (int i = array.length - 1; i >= 0; i--){**

**array[i] = 50;**

**}**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | it would generate a runtime error | | | |
|  | |  |  | | --- | --- | |  | it would result in an infinite loop | | | |
|  | |  |  | | --- | --- | |  | it would result in a compile time error | | | |
|  | |  |  | | --- | --- | |  | it would assign all of the elements of **array** the value **50** | | | |
|  | |  |  | | --- | --- | |  | it would print **50** ten times | | | |
| **Question 16**Correct on previous attempt(s) | |  | 1 / 1 point | |

Immediately after this statement executes, what best describes the result?

**int[] x = {2,3,4};**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **x** is allocated on the stack; the resulting array is also allocated on the stack. | | | |
|  | |  |  | | --- | --- | |  | **x** is a reference variable that now stores the location information for an array of 3 integer values. | | | |
|  | |  |  | | --- | --- | |  | **x** is a reference variable that now stores the location information for a two-dimensional array of integer values. | | | |
|  | |  |  | | --- | --- | |  | None of these answers. | | | |
|  | |  |  | | --- | --- | |  | **x** is a two-dimensional array of byte values. | | | |
| **Question 17**Correct on previous attempt(s) | |  | 1 / 1 point | |

What is true of the following Java code?

**int[] array = new int[10];**

**for (int i = 0; i < array.length; i--){**

**array[i] = 50;**

**}**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | it would assign all of the elements of **array** the value **50** | | | |
|  | |  |  | | --- | --- | |  | it would result in a compile time error | | | |
|  | |  |  | | --- | --- | |  | it would generate a runtime error | | | |
|  | |  |  | | --- | --- | |  | it would result in an **IndexOutOfBoundsException** being thrown | | | |
|  | |  |  | | --- | --- | |  | it would result in an infinite loop | | | |
| **Question 18**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given the following class definition:

**public class A {**

**int x;**

**public A() {**

**int x = 4;**

**}**

**public void aMethod(){**

**System.out.print(x);**

**}**

**}**

what would be printed by the following Java code?

**A a = new A();**

**a.aMethod();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **0** | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | **4** | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
|  | |  |  | | --- | --- | |  | **44** | | | |
| **Question 19**Correct on previous attempt(s) | |  | 1 / 1 point | |

Which of the following is true of an abstract class in Java?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | An abstract class can be used as the superclass in polymorphic code. | | | |
|  | |  |  | | --- | --- | |  | An abstract class cannot be extended. | | | |
|  | |  |  | | --- | --- | |  | An abstract class can only define abstract methods and cannot implement methods. | | | |
|  | |  |  | | --- | --- | |  | An abstract class is used to instantiate abstract objects. | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
| **Question 20**Correct on previous attempt(s) | |  | 1 / 1 point | |

What would be printed by executing the following Java code?

**ArrayList<int> myList;**

**myList = new ArrayList<int>();**

**System.out.print(myList.size());**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **0** | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | **10** | | | |
|  | |  |  | | --- | --- | |  | **-1** | | | |
|  | |  |  | | --- | --- | |  | Nothing, the code will not compile. | | | |
| **Question 21**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given the following class definition:

**public class A {**

**int x = 4;**

**public A() {**

**int y = 0;**

**}**

**public void aMethod(){**

**System.out.print(x);**

**}**

**}**

what would be printed by the following Java code?

**A a = new A();**

**a.aMethod();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | **44** | | | |
|  | |  |  | | --- | --- | |  | **0** | | | |
|  | |  |  | | --- | --- | |  | **4** | | | |
| **Question 22**Correct on previous attempt(s) | |  | 1 / 1 point | |

After the following Java code fragment is executed,

**try {**

**Formatter output = new Formatter("myfile");**

**} catch (Exception e){**

**//todo**

**} finally {**

**output.close();**

**}**

assuming no exception occurs, which of the following should be true?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | an empty file called **myfile.txt** is in the current working directory of the Java process | | | |
|  | |  |  | | --- | --- | |  | an empty file called **myfile** is in the user's home directory | | | |
|  | |  |  | | --- | --- | |  | the contents of **myfile** have been printed to the standard output | | | |
|  | |  |  | | --- | --- | |  | an empty file called **myfile.txt** is in the user's home directory | | | |
|  | |  |  | | --- | --- | |  | an empty file called **myfile** is in the current working directory of the Java process | | | |
| **Question 23**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given the following class definition:

**public class A {**

**int x;**

**int y;**

**public A() {**

**x = 1;**

**y = 7;**

**}**

**public void doIt(){**

**int x = 5;**

**System.out.print(x);**

**System.out.print(y);**

**}**

**}**

what would be printed by the following Java code?

**A a = new A();**

**a.doIt();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **15** | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
|  | |  |  | | --- | --- | |  | **51** | | | |
|  | |  |  | | --- | --- | |  | **17** | | | |
|  | |  |  | | --- | --- | |  | **57** | | | |
| **Question 24**Retaken | |  | 1 / 1 point | |

Which of the following is true of an abstract class in Java?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | An abstract class cannot be extended. | | | |
|  | |  |  | | --- | --- | |  | An abstract class connot be instantiated | | | |
|  | |  |  | | --- | --- | |  | An abstract class is used to instantiate abstract objects. | | | |
|  | |  |  | | --- | --- | |  | An abstract class can only define abstract methods and cannot implement methods. | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
| **Question 25**Correct on previous attempt(s) | |  | 1 / 1 point | |

What is **E** in the Java interface **List<E>**?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **E** is a type parameter that represents an interface type or a class type | | | |
|  | |  |  | | --- | --- | |  | **E** is the name of an interface or the name of a class | | | |
|  | |  |  | | --- | --- | |  | **E** is the name of a class | | | |
|  | |  |  | | --- | --- | |  | **E** is the name of an interface | | | |
| **Question 26**Correct on previous attempt(s) | |  | 1 / 1 point | |

What is true of the following Java code?

**int[] array = new int[10];**

**for (int i = array.length; i <= 0; i--){**

**array[i] = 50;**

**}**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | it would result in an array with ten elements of value **0** | | | |
|  | |  |  | | --- | --- | |  | it would result in an infinite loop | | | |
|  | |  |  | | --- | --- | |  | it would generate a runtime error | | | |
|  | |  |  | | --- | --- | |  | it would result in a compile time error | | | |
|  | |  |  | | --- | --- | |  | it would assign all of the elements of **array** the value **50** | | | |
| **Question 27**Correct on previous attempt(s) | |  | 1 / 1 point | |

To sort a Java **ArrayList**, you would use the **sort** method of which of the following?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | The **Arrays** class | | | |
|  | |  |  | | --- | --- | |  | The **Collections** class | | | |
|  | |  |  | | --- | --- | |  | The **Collection** interface | | | |
|  | |  |  | | --- | --- | |  | The **ArrayList** class | | | |
|  | |  |  | | --- | --- | |  | The **List** interface | | | |
| **Question 28**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given the following anonymous class object:

**new Bar(){**

**<java code>**

**}**

assuming **Bar** is an interface and the **<java code>** is correct, what do we know about **<java code>**?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | there is no way to know anything about it | | | |
|  | |  |  | | --- | --- | |  | it must define an anonymous method | | | |
|  | |  |  | | --- | --- | |  | it must define exactly one method | | | |
|  | |  |  | | --- | --- | |  | None of these answers. | | | |
|  | |  |  | | --- | --- | |  | it includes concrete methods to override every abstract method in the interface | | | |
| **Question 29**Correct on previous attempt(s) | |  | 1 / 1 point | |

What is true of the following Java code?

**int[10] array = new int[];**

**{**

**array[i] = 50;**

**}**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | it would result in an infinite loop | | | |
|  | |  |  | | --- | --- | |  | it would print **50** ten times | | | |
|  | |  |  | | --- | --- | |  | it would assign all of the elements of **array** the value **50** | | | |
|  | |  |  | | --- | --- | |  | it would generate a runtime error | | | |
|  | |  |  | | --- | --- | |  | it would result in a compile time error | | | |
| **Question 30**Correct on previous attempt(s) | |  | 1 / 1 point | |

One generic **Stack** class could be the basis for creating many Stack classes, for example, **Stack<Double>**, **Stack<Integer>** and **Stack<Employee>**. These classes are known as \_\_\_\_\_\_\_\_\_\_.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | subclasses | | | |
|  | |  |  | | --- | --- | |  | generic types | | | |
|  | |  |  | | --- | --- | |  | method classes | | | |
|  | |  |  | | --- | --- | |  | abstract classes | | | |
|  | |  |  | | --- | --- | |  | concrete classes | | | |
| **Question 31**Retaken | |  | 0 / 1 point | |

In the following Java code fragment:

**public class A {**

**int i;**

**Object o;**

**}**

where would **i** and **o** be stored?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Incorrect Response | |  |  | | --- | --- | |  | **O** on the stack, **i** in heap memory | | | |
| Correct Answer | |  |  | | --- | --- | |  | In heap memory | | | |
|  | |  |  | | --- | --- | |  | **i** on the stack, **o** in heap memory | | | |
|  | |  |  | | --- | --- | |  | On the stack | | | |
| **Question 32**Correct on previous attempt(s) | |  | 1 / 1 point | |

Assuming **x** and **y** are both references to objects of type **String**, which of the following Java code fragments would test whether the two strings are equal, having all the same letters in the same order:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **x == y** | | | |
|  | |  |  | | --- | --- | |  | **x equals y** | | | |
|  | |  |  | | --- | --- | |  | none of these answers | | | |
|  | |  |  | | --- | --- | |  | **x = y** | | | |
|  | |  |  | | --- | --- | |  | **x.equals(y)** | | | |
| **Question 33**Correct on previous attempt(s) | |  | 1 / 1 point | |

When working with a **Scanner** in Java, its **next** method will return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | The next integer value | | | |
|  | |  |  | | --- | --- | |  | The next delimited word of text, where the default delimiter is whitespace | | | |
|  | |  |  | | --- | --- | |  | The next Unicode character | | | |
|  | |  |  | | --- | --- | |  | The next line of input | | | |
| **Question 34**Correct on previous attempt(s) | |  | 1 / 1 point | |

When we are defining a new interface, what happens if we leave out the abstract keyword in the method definitions?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | Because all methods in an interface are abstract, it would cause a compilation error. | | | |
|  | |  |  | | --- | --- | |  | Because all methods in an interface are abstract, the abstract keyword must not appear. | | | |
|  | |  |  | | --- | --- | |  | A compilation error would result because abstract methods are not allowed in an interface. | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
| **Question 35**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given this class definition

**public class A {**

**void a() throws Exception{**

**System.out.print("A");**

**throw new Exception();**

**}**

**}**

what would be printed out by the following Java Statements?

**A a = new A();**

**try {**

**a.a();**

**} catch (Exception e){**

**System.out.print("ex");**

**}**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **exA** | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile. | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | **Aex** | | | |
|  | |  |  | | --- | --- | |  | **A** | | | |
| **Question 36**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions

**public class A {**

**public A() {**

**System.out.print("A");**

**}**

**}**

**public class B extends A {**

**public B() {**

**System.out.print("B");**

**}**

**}**

what would be printed by the following Java statements?

**A a = new A();**

**if (a instanceof B){**

**System.out.print("B");**

**else {**

**System.out.print("A");**

**}**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **ABA** | | | |
|  | |  |  | | --- | --- | |  | **AA** | | | |
|  | |  |  | | --- | --- | |  | **ABB** | | | |
|  | |  |  | | --- | --- | |  | **BB** | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
| **Question 37**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given the following Java code:

**public class A {**

**private int x = 3;**

**}**

**public class B extends A {**

**public void b(){**

**System.out.print(x);**

**}**

**}**

what would be the result of the statements

**B b = new B();**

**b.b();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **3** | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | nothing, because the code would not compile | | | |
|  | |  |  | | --- | --- | |  | **0** | | | |
| **Question 38**Retaken | |  | 1 / 1 point | |

The type-wrapper class **Integer** is immutable. What does this mean?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | It means all variables of type **Integer** are implicitly declared **final**, and cannot be changed. | | | |
|  | |  |  | | --- | --- | |  | It means an instance of the **Integer** class cannot be changed | | | |
|  | |  |  | | --- | --- | |  | It means that instances of the **Integer** class are automatically converted to **int**, or the reverse, as necessary. | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
| **Question 39**Correct on previous attempt(s) | |  | 1 / 1 point | |

Which of the following Java classes would you be most likely to use if you were implementing a program that needed to read and write file I/O with whitespace separated tokens like integers, floating point numbers, and alphanumeric strings?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **FileInputStream** and **FileOutputStream** | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | **Scanner** and **Formatter** | | | |
|  | |  |  | | --- | --- | |  | **FileReader** and **FileWriter** | | | |
|  | |  |  | | --- | --- | |  | **ObjectInputStream** and **ObjectOutputStream** | | | |
| **Question 40**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given the following Java code

**public class A {**

**protected int x = 3;**

**}**

**public class B extends A {**

**public void printIt(){**

**System.out.print(x);**

**}**

**}**

what would be the result of the statements **B b = new B();**

**b.printIt();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | nothing, because the code would not compile | | | |
|  | |  |  | | --- | --- | |  | **X** would be printed | | | |
|  | |  |  | | --- | --- | |  | **3** would be printed | | | |
|  | |  |  | | --- | --- | |  | None of the these answers | | | |
| **Question 41**Correct on previous attempt(s) | |  | 1 / 1 point | |

Which of the following is true about Objects in Java?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | Local variables implement attributes, and constructors implement behaviors. | | | |
|  | |  |  | | --- | --- | |  | Instance variables implement attributes, and constructors implement behaviors. | | | |
|  | |  |  | | --- | --- | |  | None of these answers. | | | |
|  | |  |  | | --- | --- | |  | Local variables implement attributes, and methods implement behaviors. | | | |
|  | |  |  | | --- | --- | |  | Instance variables implement attributes, and methods implement behaviors. | | | |
| **Question 42**Correct on previous attempt(s) | |  | 1 / 1 point | |

Immediately after this statement executes, what best describes the result?

**byte[][] x = {1},{2,3,4};**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **x** is a reference variable that now stores the location information for an array of 2 references to 2 arrays of byte values. | | | |
|  | |  |  | | --- | --- | |  | **x** is a two-dimensional array of byte values. | | | |
|  | |  |  | | --- | --- | |  | **x** is allocated on the stack; the resulting array is also allocated on the stack. | | | |
|  | |  |  | | --- | --- | |  | None of these answers. | | | |
|  | |  |  | | --- | --- | |  | **x** is a reference variable that now stores the location information for a two-dimensional array of byte values. | | | |
| **Question 43**Correct on previous attempt(s) | |  | 1 / 1 point | |

In the following Java code fragment:

**try {**

**int x;**

**System.out.print("A");**

**x = 3 / 0;**

**System.out.print("B");**

**} catch (ArithmeticException e){**

**System.out.print("C");**

**} catch (Exception e){**

**System.out.print("D");**

**}**

what is printed out, assuming the division by zero generates an exception?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **ABD** | | | |
|  | |  |  | | --- | --- | |  | **AC** | | | |
|  | |  |  | | --- | --- | |  | **ABC** | | | |
|  | |  |  | | --- | --- | |  | **AB** | | | |
|  | |  |  | | --- | --- | |  | **AD** | | | |
| **Question 44**Correct on previous attempt(s) | |  | 1 / 1 point | |

When we are defining a new interface, what happens if we leave out the abstract keyword in the method definitions?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | Because all methods in an interface are abstract, it would cause a compilation error. | | | |
|  | |  |  | | --- | --- | |  | A compilation error would result because abstract methods are not allowed in an interface. | | | |
|  | |  |  | | --- | --- | |  | Because all methods in an interface are abstract, the abstract keyword must not appear. | | | |
|  | |  |  | | --- | --- | |  | Because all methods in an interface are abstract, leaving out the abstract keyword is normal. | | | |
| **Question 45**Correct on previous attempt(s) | |  | 1 / 1 point | |

Which of the following examples breaks the Google Java convention for naming the type of item specified (proper capitalization, etc)?

Fragment One:

**class Raster; // class name**

Fragment Two:

**runFast(); // method name**

Fragment Three:

**float MyWidth; // variable name**

Fragment Four:

**static final int MIN\_WIDTH = 4; // constant name**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | Fragment Two | | | |
|  | |  |  | | --- | --- | |  | Fragment One | | | |
|  | |  |  | | --- | --- | |  | Fragment Four | | | |
|  | |  |  | | --- | --- | |  | Fragment Three | | | |
| **Question 46**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given the following class definition:

**public class A {**

**int x;**

**int y;**

**public A() {**

**int x = 1;**

**y = 7;**

**}**

**public void doIt(){**

**System.out.print(x);**

**System.out.print(y);**

**}**

**}**

what would be printed by the following Java code?

**A a = new A();**

**a.doIt();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
|  | |  |  | | --- | --- | |  | **51** | | | |
|  | |  |  | | --- | --- | |  | **15** | | | |
|  | |  |  | | --- | --- | |  | **07** | | | |
|  | |  |  | | --- | --- | |  | **17** | | | |
| **Question 47**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions:

**public class A {**

**public A() {**

**System.out.print("A");**

**}**

**}**

**public class B extends A {**

**public void doit() {**

**System.out.print("doit");**

**}**

**}**

what is printed out by the following Java statement:

**A b = new B();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **doitA** | | | |
|  | |  |  | | --- | --- | |  | **doit** | | | |
|  | |  |  | | --- | --- | |  | **A** | | | |
|  | |  |  | | --- | --- | |  | **Adoit** | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
| **Question 48**Correct on previous attempt(s) | |  | 1 / 1 point | |

Which of the following statements breaks the Google Java conventions for statement formatting?

Statement One:

**argv++;**

Statement Two:

**return myDisk.size();**

Statement Three:

**if (condition) {**

**statements;**

**} else if (condition) {**

**statements;**

**} else {**

**statements;**

**}**

Statement Four:

**if (condition)**

**statement;**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | Statement Four | | | |
|  | |  |  | | --- | --- | |  | Statement Two | | | |
|  | |  |  | | --- | --- | |  | Statement One | | | |
|  | |  |  | | --- | --- | |  | Statement Three | | | |
| **Question 49**Correct on previous attempt(s) | |  | 1 / 1 point | |

What is true of the following Java code?

**int[] array = new int[10];**

**for (int i = 10; i >= 0; i--){**

**array[i] = 50;**

**}**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | it would assign all of the elements of **array** the value **50** | | | |
|  | |  |  | | --- | --- | |  | it would print **50** ten times | | | |
|  | |  |  | | --- | --- | |  | it would result in an infinite loop | | | |
|  | |  |  | | --- | --- | |  | it would generate a runtime error | | | |
|  | |  |  | | --- | --- | |  | it would result in a compile time error | | | |
| **Question 50**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given the following Java code fragment:

**public void aMethod(int... args){**

**System.out.print(args[1]);**

**}**

what is the output of the statement

**aMethod(1,2,3);**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **3** | | | |
|  | |  |  | | --- | --- | |  | The statement would generate an argument mismatch error | | | |
|  | |  |  | | --- | --- | |  | **1** | | | |
|  | |  |  | | --- | --- | |  | **2** | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
| **Question 51**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given the following class definition:

**public class A {**

**int y;**

**public A(int x) {**

**int y = x;**

**}**

**public void aMethod(){**

**System.out.print(x);**

**}**

**}**

what would be printed by the following Java code?

**A a = new A(4);**

**a.aMethod();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **0** | | | |
|  | |  |  | | --- | --- | |  | **4** | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | **44** | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
| **Question 52**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions:

**public class A {**

**public A() {**

**System.out.print("A");**

**}**

**}**

**public class B extends A {**

**public B() {**

**this(4);**

**System.out.print("B");**

**}**

**public B(int x){**

**System.out.print(x);**

**}**

**}**

what is printed out by the following Java statement:

**B b = new B();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **A4B** | | | |
|  | |  |  | | --- | --- | |  | **4AB** | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
|  | |  |  | | --- | --- | |  | **AB4** | | | |
|  | |  |  | | --- | --- | |  | **AB** | | | |
| **Question 53**Retaken | |  | 0 / 1 point | |

In the following Java code fragment

**public class A {**

**int i;**

**Object o;**

**}**

where would **i** and **o** be stored?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Correct Answer | |  |  | | --- | --- | |  | In heap memory | | | |
|  | |  |  | | --- | --- | |  | On the stack | | | |
|  | |  |  | | --- | --- | |  | **i** on the stack, **o** in heap memory | | | |
| Incorrect Response | |  |  | | --- | --- | |  | **O** on the stack, **i** in heap memory | | | |
| **Question 54**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions:

**public class A {**

**public A() {**

**System.out.print("A");**

**}**

**}**

**public class B extends A {**

**public B() {**

**System.out.print("B");**

**}**

**}**

what is printed out by the following Java statement:

**A a = new B();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **B** | | | |
|  | |  |  | | --- | --- | |  | nothing | | | |
|  | |  |  | | --- | --- | |  | **BA** | | | |
|  | |  |  | | --- | --- | |  | **AB** | | | |
|  | |  |  | | --- | --- | |  | **A** | | | |
| **Question 55**Correct on previous attempt(s) | |  | 1 / 1 point | |

Which of the following lines of Java code is certainly incorrect?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **public class MyClass<T,U> {** | | | |
|  | |  |  | | --- | --- | |  | **public <T> void myMethod(T parm) {** | | | |
|  | |  |  | | --- | --- | |  | **public <T> T myMethod(T parm) {** | | | |
|  | |  |  | | --- | --- | |  | **public class MyClass<T> {** | | | |
|  | |  |  | | --- | --- | |  | **public class <T> MyClass {** | | | |
| **Question 56**Correct on previous attempt(s) | |  | 1 / 1 point | |

What would be printed by executing the following Java code?

**ArrayList<Integer> myList;**

**myList = new ArrayList<Integer>();**

**System.out.print(myList.size());**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | **-1** | | | |
|  | |  |  | | --- | --- | |  | **0** | | | |
|  | |  |  | | --- | --- | |  | Nothing, the code will not compile. | | | |
|  | |  |  | | --- | --- | |  | **10** | | | |
| **Question 57**Correct on previous attempt(s) | |  | 1 / 1 point | |

A **Comparator** in Java is

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | any object that implements the **Comparator<T>** interface and can be used to compare two objects of type **T** | | | |
|  | |  |  | | --- | --- | |  | a reference to a **Collection** object that contains ordered items | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | any object that can be ordered with respect to other objects of the same type | | | |
| **Question 58**Correct on previous attempt(s) | |  | 1 / 1 point | |

When we are defining a new interface, what happens if we leave out the abstract keyword in the method definitions?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | Because all methods in an interface are abstract, it would cause a compilation error. | | | |
|  | |  |  | | --- | --- | |  | A compilation error would result because abstract methods are not allowed in an interface. | | | |
|  | |  |  | | --- | --- | |  | Because the abstract keyword is for abstract classes, a compilation error would result. | | | |
|  | |  |  | | --- | --- | |  | Because all methods in an interface are abstract, it doesn't matter whether or not we specify the abstract keyword. | | | |
| **Question 59**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given the following class definition:

**public class A {**

**int x;**

**public A(int y) {**

**int x = y;**

**}**

**public void aMethod(){**

**System.out.print(x);**

**}**

**}**

what would be printed by the following Java code?

**A a = new A(4);**

**a.aMethod();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | **4** | | | |
|  | |  |  | | --- | --- | |  | **44** | | | |
|  | |  |  | | --- | --- | |  | **0** | | | |
| **Question 60**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions:

**public class A {**

**public A() {**

**System.out.print("A");**

**}**

**}**

**public class B extends A {**

**public B() {**

**System.out.print("B");**

**}**

**}**

which of the following statements would result in an error?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **A a = new B();** | | | |
|  | |  |  | | --- | --- | |  | **A a = new A();** | | | |
|  | |  |  | | --- | --- | |  | **A a = (A) new B();** | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | **A a = (B) new A();** | | | |
| **Question 61**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions:

**public class A {**

**public void doit() {**

**System.out.print("a");**

**}**

**}**

**public class B extends A {**

**public void doit() {**

**System.out.print("b");**

**}**

**}**

what is printed out by the following Java statements:

**B b = new A();**

**b.doit();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **ba** | | | |
|  | |  |  | | --- | --- | |  | **b** | | | |
|  | |  |  | | --- | --- | |  | **a** | | | |
|  | |  |  | | --- | --- | |  | **ab** | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
| **Question 62**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions:

**public class X {**

**int x;**

**public X(){**

**initialize();**

**printHi();**

**}**

**public void printHi(){**

**System.out.print(" X, with x = " + x);**

**}**

**public void initialize(){**

**x = 3;**

**}**

**}**

**public class Y extends X{**

**@Override**

**public void initialize(){**

**x = 99;**

**}**

**@Override**

**public void printHi(){**

**System.out.print(" Y, with x = " + super.x);**

**}**

**}**

what would be printed out by the following two statements?

**new X();**

**new Y();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **X, with x = 3 Y, with x = 3** | | | |
|  | |  |  | | --- | --- | |  | **X, with x = 99 Y, with x = 99** | | | |
|  | |  |  | | --- | --- | |  | **X, with x = 3 X, with x = 3** | | | |
|  | |  |  | | --- | --- | |  | **X, with x = 3 Y, with x = 99** | | | |
|  | |  |  | | --- | --- | |  | **Y, with x = 99 Y, with x = 99** | | | |
| **Question 63**Correct on previous attempt(s) | |  | 1 / 1 point | |

Which of the following would print all of the members of a **List<String> myLst** in order?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **while (myLst.next()) { System.out.print(myLst.next()); }** | | | |
|  | |  |  | | --- | --- | |  | **Iterator<String> i = myLst.iterator(); while (i.hasNext()) {System.out.print(i.next());}** | | | |
|  | |  |  | | --- | --- | |  | **Iterator<String> i = myLst.iterator(); while (i.next()){ System.out.print(i.next()); }** | | | |
|  | |  |  | | --- | --- | |  | **while (myLst.hasNext() { System.out.print(myLst.next); }** | | | |
| **Question 64**Correct on previous attempt(s) | |  | 1 / 1 point | |

In the following Java code fragment:

**public class MyClass {**

**public void a() {**

**int x;**

**x = 20;**

**}**

**public void b() {**

**a();**

**/\* state of x here?\*/**

**}**

**}**

what is the best description of the state of the variable **x** at the indicated point?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | **x** will not exist | | | |
|  | |  |  | | --- | --- | |  | **x** will have no value | | | |
|  | |  |  | | --- | --- | |  | **x** will have value **20** | | | |
|  | |  |  | | --- | --- | |  | **x** will be available to be garbage collected | | | |
| **Question 65**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions:

**public class A {**

**public A(int x) {**

**System.out.print(x);**

**}**

**}**

**public class B extends A {**

**public B() {**

**super(1);**

**System.out.print("B");**

**}**

**}**

what is printed out by the following Java statement:

**A a = new B();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **xB** | | | |
|  | |  |  | | --- | --- | |  | **1B** | | | |
|  | |  |  | | --- | --- | |  | **1** | | | |
|  | |  |  | | --- | --- | |  | **B** | | | |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
| **Question 66**Correct on previous attempt(s) | |  | 1 / 1 point | |

What is true of the following Java code?

**int[] array = new int[10];**

**for (int i = 0; i < array.length; i=0){**

**array[i] = 50;**

**}**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | it would result in an **IndexOutOfBoundsException** being thrown | | | |
|  | |  |  | | --- | --- | |  | it would assign all of the elements of **array** the value **50** | | | |
|  | |  |  | | --- | --- | |  | it would result in a compile time error | | | |
|  | |  |  | | --- | --- | |  | it would generate a runtime error | | | |
|  | |  |  | | --- | --- | |  | it would result in an infinite loop | | | |
| **Question 67**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given the following class definition:

**public class A {**

**int x;**

**public A(int x) {**

**this.x = x;**

**}**

**public void aMethod(){**

**System.out.print(x);**

**}**

**}**

what would be printed by the following Java code?

**A a = new A(4);**

**a.aMethod();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | nothing, the code won't compile | | | |
|  | |  |  | | --- | --- | |  | **0** | | | |
|  | |  |  | | --- | --- | |  | **44** | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | **4** | | | |
| **Question 68**Correct on previous attempt(s) | |  | 1 / 1 point | |

Given these class definitions:

**public class A {**

**public void a() {**

**System.out.print("A");**

**}**

**}**

**public class B extends A {**

**public B() {**

**System.out.print("B");**

**}**

**}**

what is printed out by the following Java statement:

**A a = new B();**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **AB** | | | |
|  | |  |  | | --- | --- | |  | **BA** | | | |
|  | |  |  | | --- | --- | |  | nothing | | | |
|  | |  |  | | --- | --- | |  | **B** | | | |
|  | |  |  | | --- | --- | |  | **A** | | | |
| **Question 69**Correct on previous attempt(s) | |  | 1 / 1 point | |

In the following Java code fragment:

**public class MyClass {**

**int x;**

**public void a() {**

**x = 17;**

**}**

**public void b() {**

**a();**

**/\* state of x here?\*/**

**}**

**}**

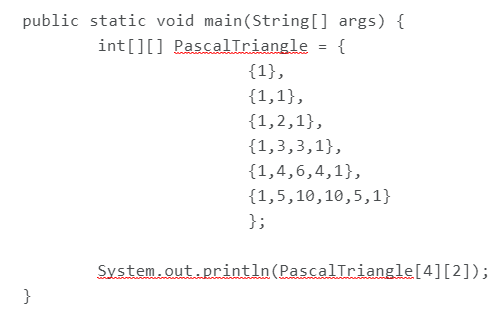
what is the best description of the state of the variable **x** at the indicated point?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | **x** will have value **17** | | | |
|  | |  |  | | --- | --- | |  | **x** will be available to be garbage collected | | | |
|  | |  |  | | --- | --- | |  | None of these answers | | | |
|  | |  |  | | --- | --- | |  | **x** will not exist | | | |
|  | |  |  | | --- | --- | |  | **x** will have no value | | | |
| **Question 70**Correct on previous attempt(s) | |  | 1 / 1 point | |

Which statement is false about Java Generics?

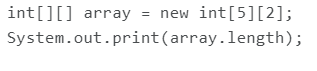
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | |  |  | | --- | --- | |  | When declaring a generic method, the type parameter section is placed before the return type of the method. | | | |
|  | | | |  |  | | --- | --- | |  | Each Diamond Operator (type parameter section) contains at most one type parameter. | | | |
|  | | | |  |  | | --- | --- | |  | A type parameter is an identifier that specifies a generic type name. | | | |
|  | | | |  |  | | --- | --- | |  | Type parameters can represent only reference types. | | | |
|  |  |
| **Question 1** | | | |  | 1 / 1 point | |

What is the output of the following?:



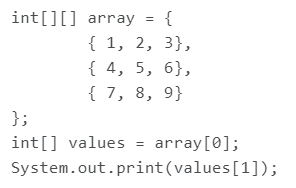
|  |  |
| --- | --- |
|  | nothing, the code won’t compile |
|  | 6 |
|  | 4 |
|  | 3 |
|  | runtime exception |
| **Question 2** | |  | 1 / 1 point |

What is result of the following Java code?



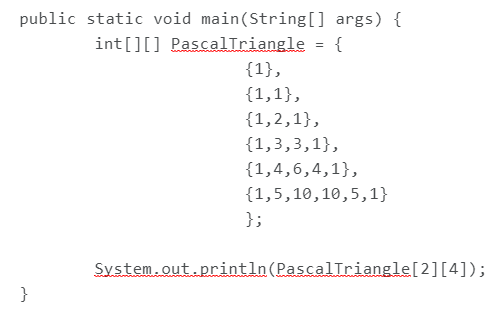
|  |  |
| --- | --- |
|  | 2 |
|  | 5 |
|  | 10 |
|  | 0 |
|  | None of the above |
| **Question 3** | |  | 1 / 1 point |

What is result of the following Java code?



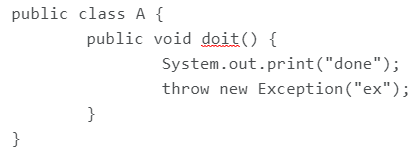
|  |  |
| --- | --- |
|  | compiler error |
|  | 5 |
|  | ArrayIndexOutOfBoundsException |
|  | 2 |
|  | 4 |
| **Question 4** | |  | 1 / 1 point |

What is the output of the following?:



|  |  |
| --- | --- |
|  | nothing, the code won’t compile |
|  | 3 |
|  | 4 |
|  | 6 |
|  | runtime exception |
| **Question 5** | |  | 1 / 1 point |

Given this class definition



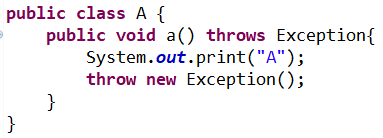
what would be printed by the following Java statements?

A a = new A();

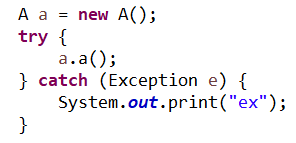
a.doit();

|  |  |
| --- | --- |
|  | ex |
|  | nothing, the code won’t compile. |
|  | done |
|  | doneex |
|  | None of these answers |
| **Question 6** | |  | 1 / 1 point |

Given this class definition

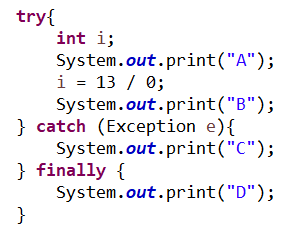


what would be printed out by the following Java Statements?



|  |  |
| --- | --- |
|  | A |
|  | None of these answers |
|  | nothing, the code won’t compile. |
|  | Aex |
|  | exA |
| **Question 7** | |  | 1 / 1 point |

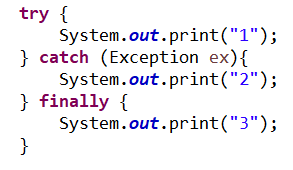
In the following Java code fragment



what is printed out, assuming the division by zero generates an exception?

|  |  |
| --- | --- |
|  | AD |
|  | CD |
|  | ADCB |
|  | ACD |
|  | AC |
| **Question 8** | |  | 1 / 1 point |

In the following Java code fragment:



what is printed out assuming no exception occurs?

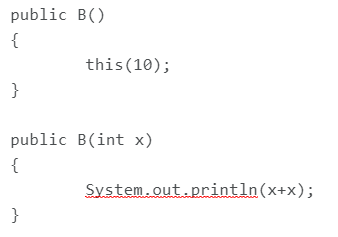
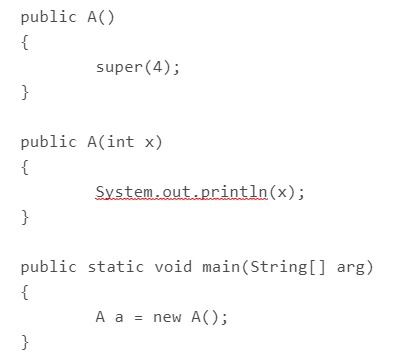
|  |  |
| --- | --- |
|  | 3 |
|  | 12 |
|  | 1 |
|  | 2 |
|  | 13 |
| **Question 9** | |  | 0 / 1 point |

Which of the following statement is most true:

|  |  |  |  |
| --- | --- | --- | --- |
| Correct Answer | |  |  | | --- | --- | | a) | The **throws** keyword is used in the signature of a method to indicate that this method might throw an exception | |
|  | |  |  | | --- | --- | | b) | The **throwException** keyword is used to explicitly throw an exception from a method | |
|  | |  |  | | --- | --- | | c) | All exceptions that are not “Error” **must** be handled | |
|  | |  |  | | --- | --- | | d) | Catch blocks **must** be placed from the general to the specific | |
| Incorrect Response | |  |  | | --- | --- | | e) | a) and b) above | |
| **Question 10** | |  | 1 / 1 point |

Which of these is the correct way of calling a constructor having no parameters, of superclass A by subclass B?

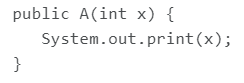
|  |  |
| --- | --- |
|  | super(void); |
|  | super(); |
|  | superclass.(); |
|  | super.A(); |
| **Question 11** | |  | 1 / 1 point |

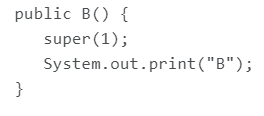
You have a class named B with the following in it:   
  
  
You have a class named A that extends class B, and has the following in it:  


What will be the output?

|  |  |
| --- | --- |
|  | 20 |
|  | 4 |
|  | Compile error |
|  | 8 |
|  | 10 |
| **Question 12** | |  | 1 / 1 point |

You have a class named A with the following in it:



You have a class named B that extends class A, and has the following in it:  


What is printed out by the following Java statement:

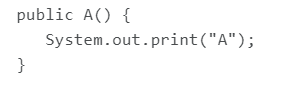
A a = new B();

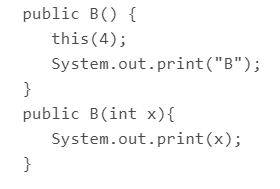
|  |  |
| --- | --- |
|  | None of these |
|  | xB |
|  | B |
|  | 1B |
|  | 1 |
| **Question 13** | |  | 1 / 1 point |

The default constructor of a subclass always calls the default constructor of it's superclass

|  |  |  |
| --- | --- | --- |
|  |  | True |
|  |  | False |
| **Question 14** | | |  | 1 / 1 point |

You have a class named A with the following in it:



You have a class named B that extends class A, and has the following in it:  


What is printed out by the following Java statement:

B b = new B();

|  |  |
| --- | --- |
|  | None of these |
|  | AB4 |
|  | A4B |
|  | 4AB |
|  | AB |
| **Question 15** | |  | 1 / 1 point |

Which of those **does not** allow duplicate elements?

|  |  |
| --- | --- |
|  | Collections |
|  | Set |
|  | Collection |
|  | List |
| **Question 16** | |  | 1 / 1 point |

To sort a Java ArrayList, you would use the sort method of which of the following?

|  |  |
| --- | --- |
|  | The Collections class |
|  | The ArrayList class |
|  | The Collection interface |
|  | The List interface |
|  | The Arrays class |
| **Question 17** | |  | 1 / 1 point |

Which of these helps ***efficiently***remove a element at a specific position in a collection?

|  |  |
| --- | --- |
|  | ArrayList |
|  | Vector |
|  | List |
|  | LinkedList |
| **Question 18** | |  | 1 / 1 point |

The root interface of Java collection framework hierarchy is:

|  |  |
| --- | --- |
|  | Collections |
|  | Collection |
|  | Set |
|  | Root |
|  | List |
| **Question 19** | |  | 1 / 1 point |

Which of these helps ***efficiently***insert elements at a specific position in a collection?

|  |  |
| --- | --- |
|  | Vector |
|  | ArrayList |
|  | List |
|  | LinkedList |
| **Question 20** | |  | 1 / 1 point |

What Java keyword is used to inherit an interface?

|  |  |
| --- | --- |
|  | inherits |
|  | overloads |
|  | extends |
|  | implements |
|  | overrides |
| **Question 21** | |  | 1 / 1 point |

Which of the following is **false**:

|  |  |
| --- | --- |
|  | An abstract class must have at least one abstract method. |
|  | To use you need to extend an abstract class to a concrete class. |
|  | You can extend a concrete class to an abstract class |
|  | An abstract class can have any constructors |
| **Question 22** | |  | 1 / 1 point |

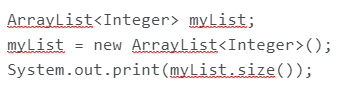
Which of the following is **false**:

|  |  |
| --- | --- |
|  | Variables declared in a Java interface is by default final. An abstract class may contain non-final variables. |
|  | An interface can extend another Java interface only, an abstract class can extend another Java class and implement multiple Java interfaces. |
|  | A Java class can implement multiple interfaces but it can extend only one abstract class. |
|  | A Java interface are implicitly abstract and cannot have implementations. A Java abstract class can have instance methods that implements a default behavior. |
|  | Members of an abstract class are public by default. An interface can have private, protected, etc.. |
| **Question 23** | |  | 0 / 1 point |

Which of the following statements about Abstract Classes and Interfaces are true?

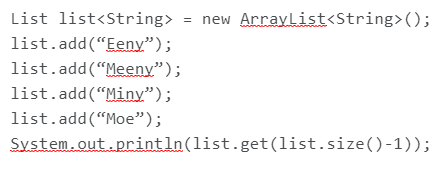
|  |  |
| --- | --- |
|  | all of these |
|  | a class can extend more than one abstract class, but a class can implement only one interface |
|  | all attributes in an abstract class, if any, must be constants (i.e. public static final) |
|  | an interface doesn’t include implementation (method bodies) for any of its method definitions |
| **Question 24** | |  | 0 / 1 point |

What would be printed by executing the following Java code (assume that all required import statements are included)?



|  |  |
| --- | --- |
|  | 10 |
|  | Nothing, the code will not compile |
|  | 0 |
|  | -1 |
|  | None of these answers |
| **Question 25** | |  | 1 / 1 point |

What is the output of the following code segment:



|  |  |
| --- | --- |
|  | Eeny |
|  | Meeny |
|  | Miny |
|  | Moe |
|  | Throw an exception |
| **Question 26** | |  | 1 / 1 point |

Which symbol is association?

|  |  |
| --- | --- |
|  | Arrow with dotted line |
|  | None-triangular Arrow |
|  | Empty diamond |
|  | Filled in diamond |
|  | Triagular Arrow |
| **Question 27** | |  | 1 / 1 point |

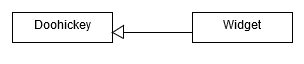
Which symbol is realization ?

|  |  |
| --- | --- |
|  | Triagular Arrow |
|  | None-triangular Arrow |
|  | Empty diamond |
|  | Filled in diamond |
|  | Arrow with dotted line |
| **Question 28** | |  | 1 / 1 point |

The statement “class DoHickey extends ThingaBob” means?

|  |  |
| --- | --- |
|  | ThingaBob is-a DoHickey |
|  | all of the others |
|  | ThingaBob inherits from DoHickey |
|  | ThingaBob is a generalization of DoHickey |
| **Question 29** | |  | 1 / 1 point |

What does the following mean?

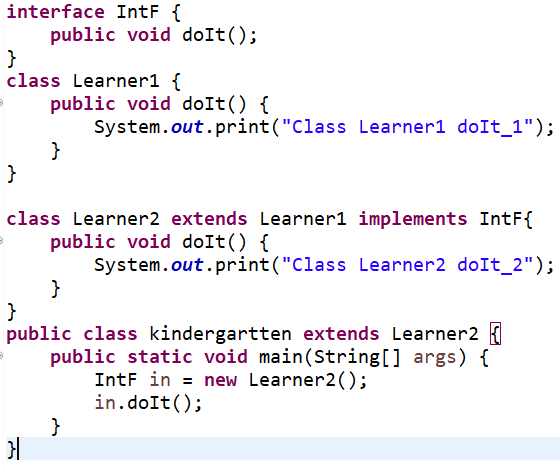


|  |  |
| --- | --- |
|  | A Widget use a Doohickey |
|  | A Widget has a Doohickey |
|  | A Doohickey is a Widget |
|  | A Doohickey has a Widget |
|  | A Widget is a Doohickey |
| **Question 30** | |  | 4 / 4 points |

Match access levels to UML symbol

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  | \_\_2\_\_ |  | + | |  | \_\_1\_\_ |  | - | |  | \_\_3\_\_ |  | # | |  | \_\_4\_\_ |  | ~ | |  | |  |  | | --- | --- | | **1**. | private | | **2**. | public | | **3**. | protected | | **4**. | default (no modifier) | | |
| **Question 31** | |  | 1 / 1 point | |

What is the output of the program below?

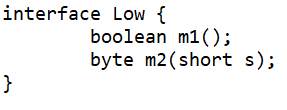


|  |  |
| --- | --- |
|  | Class Learner2 doIt\_2 |
|  | Class Learner1 doIt\_1 |
|  | none of these |
|  | gets compiled fully but prints nothing |
|  | Compilation error |
| **Question 32** | |  | 1 / 1 point |

Choose the class definition that defines a legal abstract class.

|  |  |
| --- | --- |
|  | public class abstract D { abstract void incomplete(); } |
|  | class D { abstract void incomplete() { }; } |
|  | abstract class D { abstract void incomplete(); } |
|  | class D { abstract void incomplete(); } |
| **Question 33** | |  | 1 / 1 point |

Based on this code:

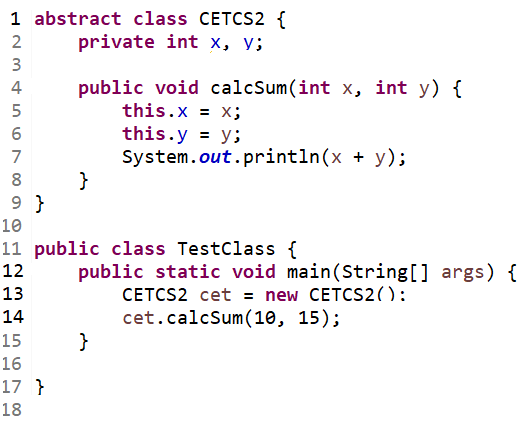


which of the two code fragments shown below will compile?

1. interface High implements Low{}  
2. abstract class MidPoint extends Low  
    { public boolean m1(){ return true;}  
3. abstract class MidPoint implements Low{}  
4. abstract class MidPoint implements Low  
    {public boolean m1(){ return (8 > 5);}  
5. abstract class MidPoint implements Low  
    {protected boolean m1() { return (3 < 5);}

|  |  |
| --- | --- |
|  | 3 and 4 |
|  | 1 and 2 |
|  | 2 and 3 |
|  | 1 and 3 |
|  | 4 and 5 |
| **Question 34** | |  | 0 / 1 point |

What will happen when this program is compiled?



|  |  |
| --- | --- |
|  | run successfully and print 25 |
|  | runtime error due line 2 |
|  | compilation error due to line 13 |
|  | compilation error due to line 7 |
|  | None of these |
| **Question 35** | |  | 1 / 1 point |

Suppose B is an abstract class, C is a concrete subclass of B, and both B and C have a no-arg constructor. Which of the following is correct?

1. B  b = new B();

2. B  b = new C ();

3. C  c = new B();

4. C  c = new C();

|  |  |
| --- | --- |
|  | 2 and 3 |
|  | 3 and 4 |
|  | 2 and 4 |
|  | 1 and 3 |
| **Question 36** | |  | 1 / 1 point |

When working with a Scanner in Java, its next method will return

|  |  |
| --- | --- |
|  | The next Unicode character |
|  | The next line of input |
|  | The next integer value |
|  | None of these answers |
|  | The next word of input |
| **Question 37** | |  | 1 / 1 point |

What is true of class variables in Java that are declared to be static?

|  |  |
| --- | --- |
|  | They are a property of every object that is an instance of the class |
|  | They are associated with the class in which they are declared as opposed to objects that are instances of the class |
|  | They are implemented on the stack |
|  | They are immutable and cannot be changed |
| **Question 38** | |  | 1 / 1 point |

Which keyword is used to import a package from the Java API library?

|  |  |
| --- | --- |
|  | import |
|  | getlib |
|  | package |
|  | lib |
| **Question 39** | |  | 1 / 1 point |

Which of the following statements is **true**:

|  |  |
| --- | --- |
|  | A static variable cannot be changed |
|  | A multidimensional array can have more than three dimensions |
|  | A package is intended to group related objects |
|  | Polymorphism refers to the bundling of data with the methods that operate on that data |
|  | A List can contain both primitives and reference types |
| **Question 40** | |  | 1 / 1 point |

The statement that opens file "newTrans.ser" for output using ObjectOutputStream variable outTrans to wrap a FileOutputStream object is:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | ObjectOutputStream outTrans = new ObjectOutputStream (new FileOutputStream ("newtrans.ser")); | |
|  | |  |  | | --- | --- | |  | ObjectOutputStream outTrans = new ObjectInputStream (new FileInputStream ("newtrans.ser")); | |
|  | |  |  | | --- | --- | |  | ObjectInputStream outTrans = new ObjectOutputStream (new FileOutputStream ("newtrans.ser")); | |
|  | |  |  | | --- | --- | |  | none of the above | |
| **Question 41** | |  | 1 / 1 point |

An absolute path contains all the directories, starting with the root directory, that lead to a specific file or directory.

|  |  |  |
| --- | --- | --- |
|  |  | 1) True |
|  |  | 2) False |
| **Question 42** | | |  | 0 / 1 point |

Which of the following classes is *not* used for file input?

|  |  |
| --- | --- |
|  | FileReader |
|  | FileInputStream |
|  | Formatter |
|  | ObjectInputStream |
| **Question 43** | |  | 1 / 1 point |

The following code will execute successfully: (assuming the file payables.txt contains as it's first data 150.00): Scanner inPayable = new Scanner (new File ("payables.txt"); double amount = inPayable.nextDouble();

|  |  |  |
| --- | --- | --- |
|  |  | 1) True |
|  |  | 2) False |
| **Question 44** | | |  | 1 / 1 point |

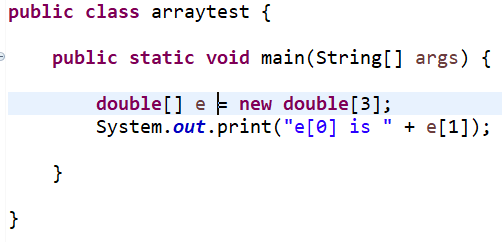
Binary files are human readable in a text editor.

|  |  |  |
| --- | --- | --- |
|  |  | 1) True |
|  |  | 2) False |
| **Question 45** | | |  | 1 / 1 point |

Select the valid statement among these

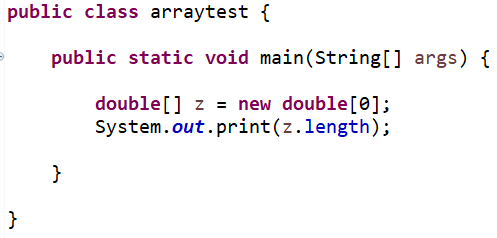
|  |  |
| --- | --- |
|  | char[] b = new char(6); |
|  | char[] b = new char[]; |
|  | char[] b = new char(); |
|  | char[] b = new char[7]; |
| **Question 46** | |  | 1 / 1 point |

What is does the program below output?



|  |  |
| --- | --- |
|  | a runtime error caused by e [1] in the output statement |
|  | a compile time error since an incorrect array size was specified in the array declaration |
|  | a runtime error caused by failing to initialize array elements |
|  | The program compiles and runs fine and displaying e[0] is 0.0 as output |
| **Question 47** | |  | 1 / 1 point |

What is the output of the following program code?



|  |  |
| --- | --- |
|  | Compile-time error, should use z.length() instead of z.length |
|  | none of the above and or below |
|  | 0 |
|  | compile-time error, arrays cannot be initialized to zero |
|  |  |
|  | |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 1 / 1 point |

In MVC (Model-View-Controller) pattern, which block of application deals with main data/application.

|  |  |  |  |
| --- | --- | --- | --- |
|  | view | | |
|  | model | | |
|  | controller | | |
| **Question 2** | |  | 1 / 1 point | |

Which category of design patterns characterize the ways in which classes or objects  interact and distribute responsibility.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Structural | | |
|  | Behavioural | | |
|  | Creational | | |
| **Question 3** | |  | 1 / 1 point | |

Which design pattern ensures that there exists only one instance of a class within the system.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Factory | | |
|  | Singleton | | |
|  | Decorator | | |
|  | MVC | | |
| **Question 4** | |  | 1 / 1 point | |

Which design pattern is also known as Wrapper with the intention to attach additional responsibilities to an object dynamically.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Singleton | | |
|  | Decorator | | |
|  | Factory | | |
|  | Facade | | |
| **Question 5** | |  | 1 / 1 point | |

In MVC (Model-View-Controller) pattern, which block of application deals with the presentation of information.

|  |  |  |  |
| --- | --- | --- | --- |
|  | model | | |
|  | controller | | |
|  | view | | |
| **Question 6** | |  | 1 / 1 point | |

In MVC (Model-View-Controller) pattern, which block of application deals with

controlling the logic and behavior of the application.

|  |  |  |  |
| --- | --- | --- | --- |
|  | view | | |
|  | model | | |
|  | controller | | |
| **Question 7** | |  | 1 / 1 point | |

Which design pattern provides a way to access the elements of an aggregate object sequentially without  exposing its underlying representation.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Iterator | | |
|  | Singleton | | |
|  | Decorator | | |
|  | Factory | | |
| **Question 8** | |  | 1 / 1 point | |

Which pattern returns an instance of one of several possible classes depending on the  data provided to it.

|  |  |
| --- | --- |
|  | Decorator |
|  | Singleton |
|  | Factory |
|  | Iterator |

|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 0 / 1 point |

JUnit is primarily used for:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Acceptance testing | | |
|  | Unit testing | | |
|  | Integration testing | | |
|  | All of the others | | |
|  | System testing | | |
| **Question 2** | |  | 1 / 1 point | |

Which of the following is correct about JUnit?

|  |  |  |  |
| --- | --- | --- | --- |
|  | JUnit tests can be organized into test suites containing test cases and even other test suites. | | |
|  | JUnit shows test progress in a bar that is green if test is going fine and it turns red when a test fails in eclipse. | | |
|  | Both of the above | | |
|  | None of the above. | | |
| **Question 3** | |  | 1 / 1 point | |

To sort a Java ArrayList, you would use the sort method of which of the following?

|  |  |  |  |
| --- | --- | --- | --- |
|  | The Collections class | | |
|  | The Collection interface | | |
|  | The Arrays class | | |
|  | The List interface | | |
|  | The ArrayList class | | |
| **Question 4** | |  | 1 / 1 point | |

java.util.Collection is a:

|  |  |  |  |
| --- | --- | --- | --- |
|  | None of these | | |
|  | Interface | | |
|  | Class | | |
|  | Object | | |
| **Question 5** | |  | 1 / 1 point | |

Which of these helps ***efficiently*** remove a element at a specific position in a collection?

|  |  |  |  |
| --- | --- | --- | --- |
|  | LinkedList | | |
|  | Vector | | |
|  | ArrayList | | |
|  | List | | |
| **Question 6** | |  | 1 / 1 point | |

Which of the following is an “unchecked exception”?

|  |  |  |  |
| --- | --- | --- | --- |
|  | RuntimeException | | |
|  | Error | | |
|  | IOException | | |
|  | All of the above | | |
|  | None of the above | | |
| **Question 7** | |  | 1 / 1 point | |

When should a program *explicitly* use the **this** reference?

|  |  |  |  |
| --- | --- | --- | --- |
|  | Accessing a private variable. | | |
|  | Accessing a field that has the same name as a local variable. | | |
|  | Accessing a public variable | | |
|  | Accessing a local variable | | |
| **Question 8** | |  | 0 / 1 point | |

What is true of class variables in Java that are declared to be static?

|  |  |  |  |
| --- | --- | --- | --- |
|  | They are implemented on the stack | | |
|  | They are a property of every object that is an instance of the class | | |
|  | There is only one copy of a static class variable, shared by all instances | | |
|  | They are immutable and cannot be changed | | |
| **Question 9** | |  | 1 / 1 point | |

Overriding a method differs from overloading a method because:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Overloaded methods have the same signature. | | |
|  | Overridden methods have the same signature. | | |
|  | Both of the above. | | |
|  | Neither of the above. | | |
| **Question 10** | |  | 1 / 1 point | |

When working with a Scanner in Java, its next method will return

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | The next integer value |
|  | | | The next Unicode character |
|  | | | None of these answers |
|  | | | The next word of input |
|  | | | The next line of input |
|  |  |
|  | |

Given this class definition

Text, letter

Description automatically generated

what would be printed out by the following Java Statements?

Text, letter

Description automatically generated

|  |  |
| --- | --- |
|  | nothing, the code won’t compile. |
|  | A |
|  | None of these answers |
|  | exA |
|  | Aex |
| **Question 2** | |  | 1 / 1 point |

Which of the following statement is most true:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | | a) | The **throws** keyword is used in the signature of a method to indicate that this method might throw an exception | |
|  | |  |  | | --- | --- | | b) | The **throwException** keyword is used to explicitly throw an exception from a method | |
|  | |  |  | | --- | --- | | c) | All exceptions that are not “Error” **must** be handled | |
|  | |  |  | | --- | --- | | d) | Catch blocks **must** be placed from the general to the specific | |
|  | |  |  | | --- | --- | | e) | a) and b) above | |
| **Question 3** | |  | 1 / 1 point |

In the catch block below, what is arithmeticException?

catch ( ArithmeticException arithmeticException )

{

System.err.printf( arithmeticException );

}

|  |  |
| --- | --- |
|  | An exception handler |
|  | A finally block |
|  | The name of catch block’s exception parameter |
|  | The type of the exception being caught |
| **Question 4** | |  | 1 / 1 point |

Given this class definition

Graphical user interface, text, application

Description automatically generated

what would be printed by the following Java statements?

A a = new A();

a.doit();

|  |  |
| --- | --- |
|  | None of these answers |
|  | ex |
|  | doneex |
|  | nothing, the code won’t compile. |
|  | done |
| **Question 5** | |  | 1 / 1 point |

In the following Java code fragment:

Text

Description automatically generated

what is printed out assuming no exception occurs?

|  |  |
| --- | --- |
|  | 3 |
|  | 12 |
|  | 1 |
|  | 2 |
|  | 13 |
| **Question 6** | |  | 1 / 1 point |

What is result of the following Java code?

Text

Description automatically generated with medium confidence

|  |  |
| --- | --- |
|  | 2 |
|  | 5 |
|  | 10 |
|  | 0 |
|  | None of the above |
| **Question 7** | |  | 1 / 1 point |

What is the output of the following?:

Text

Description automatically generated

|  |  |
| --- | --- |
|  | 4 |
|  | 6 |
|  | nothing, the code won’t compile |
|  | runtime exception |
|  | 3 |
| **Question 8** | |  | 1 / 1 point |

What is result of the following Java code?

Text

Description automatically generated with medium confidence

|  |  |
| --- | --- |
|  | 2 |
|  | 5 |
|  | 10 |
|  | 0 |
|  | None of the above |
| **Question 9** | |  | 1 / 1 point |

You have an array named array of type int and of size 10. What is true of the following Java code?

A picture containing chart

Description automatically generated

|  |  |
| --- | --- |
|  | it would generate a runtime error |
|  | it would assign all of the elements of array the value 50 |
|  | it would assign all of the elements of array the value 0 |
|  | it would result in a compile-time error |
|  | it would result in an IndexOutOfBoundsException being thrown |
| **Question 10** | |  | 1 / 1 point |

To sort a Java ArrayList, you would use the sort method of which of the following?

|  |  |
| --- | --- |
|  | The List interface |
|  | The ArrayList class |
|  | The Arrays class |
|  | The Collection interface |
|  | The Collections class |
| **Question 11** | |  | 0 / 1 point |

LinkedList implements?

|  |  |  |  |
| --- | --- | --- | --- |
| Incorrect Response | |  |  | | --- | --- | | a) | List | |
|  | |  |  | | --- | --- | | b) | Set | |
|  | |  |  | | --- | --- | | c) | Collection | |
| Correct Answer | |  |  | | --- | --- | | d) | a) and c) | |
| **Question 12** | |  | 1 / 1 point |

Which of those allows duplicate elements?

|  |  |
| --- | --- |
|  | List |
|  | Collections |
|  | Set |
|  | Collection |
| **Question 13** | |  | 1 / 1 point |

LinkedList implements?

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | | a) | List | |
|  | |  |  | | --- | --- | | b) | Set | |
|  | |  |  | | --- | --- | | c) | Collections | |
|  | |  |  | | --- | --- | | d) | a) and c) | |
| **Question 14** | |  | 1 / 1 point |

Which of those **does not** allow duplicate elements?

|  |  |
| --- | --- |
|  | Collections |
|  | Collection |
|  | Set |
|  | List |
| **Question 15** | |  | 1 / 1 point |

ArrayList is an example of :

|  |  |
| --- | --- |
|  | generic |
|  | Collection |
|  | Data structure |
|  | None of the above |
|  | All of the above |
| **Question 16** | |  | 1 / 1 point |

What is the output of the following code segment:

Graphical user interface, text, application

Description automatically generated

|  |  |
| --- | --- |
|  | Eeny |
|  | Meeny |
|  | Miny |
|  | Moe |
|  | Throw an exception |
| **Question 17** | |  | 1 / 1 point |

Which of these statements is **false**:

|  |  |
| --- | --- |
|  | We cannot use this() and super() in the same constructor |
|  | this() and super() must be the first statement in the constructor |
|  | None of these |
|  | this() can call same class constructor only |
|  | super() can call immediate super class constructor only |
| **Question 18** | |  | 1 / 1 point |

You have a class named A with the following in it:

A picture containing logo

Description automatically generated

You have a class named B that extends class A, and has the following in it:  
A picture containing company name

Description automatically generated

What is printed out by the following Java statements:

A a = new B();

a.doIt();

|  |  |
| --- | --- |
|  | b |
|  | a |
|  | None of these |
|  | ab |
|  | ba |
| **Question 19** | |  | 1 / 1 point |

You have a class named A with the following in it:

A picture containing text

Description automatically generated

You have a class named B that extends class A, and has the following in it:  
Text

Description automatically generated

What is printed out by the following Java statement:

B b = new B();

|  |  |
| --- | --- |
|  | A4B |
|  | 4AB |
|  | AB4 |
|  | AB |
|  | None of these |
| **Question 20** | |  | 1 / 1 point |

You have a class named B with the following in it:   
Graphical user interface, text, application, email

Description automatically generated  
  
You have a class named A that extends class B, and has the following in it:  
Graphical user interface, text, application

Description automatically generated

What will be the output?

|  |  |
| --- | --- |
|  | 10 |
|  | 8 |
|  | Compile error |
|  | 4 |
|  | 20 |
| **Question 21** | |  | 1 / 1 point |

You have a class named A with the following in it:

A picture containing text

Description automatically generated

You have a class named B that extends class A, and has the following in it:  
A picture containing text

Description automatically generated

Which of the following statements would result in a runtime error?

|  |  |
| --- | --- |
|  | A a = new B(); |
|  | A a = (B) new A(); |
|  | A a = new A(); |
|  | A a = (A) new B(); |
| **Question 22** | |  | 1 / 1 point |

When should a program *explicitly* use the **this** reference?

|  |  |
| --- | --- |
|  | Accessing a public variable |
|  | Accessing a private variable. |
|  | Accessing a local variable |
|  | Accessing a field that has the same name as a local variable. |
| **Question 23** | |  | 0 / 1 point |

If a parent class has a public method called “MyMethod” then the child classes:

|  |  |
| --- | --- |
|  | Must not implement MyMethod |
|  | Must implement MyMethod |
| Incorrect Response | May implement MyMethod but only if the parameters match |
| Correct Answer | May implement MyMethod and the parameters may or may not match |
|  | May implement MyMethod but only if the parameters do not match |
| **Question 24** | |  | 1 / 1 point |

Which of the following is true about **Objects** in Java?

|  |  |
| --- | --- |
|  | None of these answers. |
|  | Local variables implement attributes, and methods implement behaviors. |
|  | Instance variables implement attributes, and constructors implement behaviors. |
|  | Local variables implement attributes, and constructors implement behaviors. |
|  | Instance variables implement attributes, and methods implement behaviors. |
| **Question 25** | |  | 1 / 1 point |

What is true of class variables in Java that are declared to be static?

|  |  |
| --- | --- |
|  | They are associated with the class in which they are declared as opposed to objects that are instances of the class |
|  | They are immutable and cannot be changed |
|  | They are a property of every object that is an instance of the class |
|  | They are implemented on the stack |
| **Question 26** | |  | 1 / 1 point |

The difference between aggregation and composition is:

|  |  |
| --- | --- |
|  | There are the same thing |
|  | They are not related |
|  | Aggregation is a stronger form of composition |
|  | Composition is a stronger form of aggregation |
| **Question 27** | |  | 1 / 1 point |

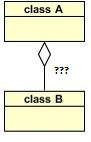
Which symbol is generalization ?

|  |  |
| --- | --- |
|  | Arrow with dotted line |
|  | None-triangular Arrow |
|  | Triagular Arrow |
|  | Filled in diamond |
|  | Empty diamond |
| **Question 28** | |  | 1 / 1 point |

Which symbol is realization ?

|  |  |
| --- | --- |
|  | Filled in diamond |
|  | Triagular Arrow |
|  | Arrow with dotted line |
|  | None-triangular Arrow |
|  | Empty diamond |
| **Question 29** | |  | 1 / 1 point |

What is the best word for the relationship?



|  |  |
| --- | --- |
|  | uses |
|  | is-a |
|  | owns |
|  | realizes |
|  | contains |
| **Question 30** | |  | 1 / 1 point |

An absolute path contains all the directories, starting with the root directory, that lead to a specific file or directory.

|  |  |  |
| --- | --- | --- |
|  |  | 1) True |
|  |  | 2) False |
| **Question 31** | | |  | 1 / 1 point |

If an absolute path of a file is NOT specified when declaring a file object and connecting to that file to read from it, the file must exist be in the project directory.

|  |  |  |
| --- | --- | --- |
|  |  | 1) True |
|  |  | 2) False |
| **Question 32** | | |  | 1 / 1 point |

The following code will execute successfully: (assuming the file payables.txt contains as it's first data 150.00): Scanner inPayable = new Scanner (new File ("payables.txt"); double amount = inPayable.nextDouble();

|  |  |  |
| --- | --- | --- |
|  |  | 1) True |
|  |  | 2) False |
| **Question 33** | | |  | 1 / 1 point |

The statement(s) that reads a record from the file "trans.ser" into an object of class TransactionRecord (using the ObjectInputStream variable inTrans which has been declared successfully already) is:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | TransactionRecord trans = inTrans.readObject(); | |
|  | |  |  | | --- | --- | |  | TransactionRecord trans = (TransactionRecord) inTrans.readObject(); | |
|  | |  |  | | --- | --- | |  | TransactionRecord trans = (TransactionRecord) inTrans.writeObject(); | |
|  | |  |  | | --- | --- | |  | none of the above. | |
| **Question 34** | |  | 1 / 1 point |

The statement that outputs a record of type TransactionRecord (stored in object transactionRecord) to the file "newTrans.ser" (which has been connected to the object outTrans of type ObjectOutputStream successfully already) is:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | outTrans.writeObject(); | |
|  | |  |  | | --- | --- | |  | outTrans = writeObject(); | |
|  | |  |  | | --- | --- | |  | outTrans.writeObject("newTrans.ser"); | |
|  | |  |  | | --- | --- | |  | none of the above. | |
| [View Feedback](javascript://) | |

|  |  |  |
| --- | --- | --- |
| **Question 35** |  | 0 / 1 point |

Which of the following statements about Abstract Classes and Interfaces are true?

|  |  |
| --- | --- |
| Incorrect Response | all of these |
|  | all attributes in an abstract class, if any, must be constants (i.e. public static final) |
|  | a class can extend more than one abstract class, but a class can implement only one interface |
| Correct Answer | an interface doesn’t include implementation (method bodies) for any of its method definitions |
| **Question 36** | |  | 1 / 1 point |

Which of the following is **false**:

|  |  |
| --- | --- |
|  | To use you need to extend an abstract class to a concrete class. |
|  | You can extend a concrete class to an abstract class |
|  | An abstract class must have at least one abstract method. |
|  | An abstract class can have any constructors |
| **Question 37** | |  | 1 / 1 point |

Java Interfaces are useful because:

|  |  |
| --- | --- |
|  | all of these |
|  | they form a contract specifying what behaviour is provided by a class that implements the interface without forcing that behaviour to be implemented in a certain way |
|  | they can help coordinate between two separate groups of programmers when one group needs to use methods that the other group implements |
|  | they can help define an Application Programming Interface (API) |
| **Question 38** | |  | 1 / 1 point |

What Java keyword is used to inherit an interface?

|  |  |
| --- | --- |
|  | overrides |
|  | inherits |
|  | implements |
|  | extends |
|  | overloads |
| **Question 39** | |  | 1 / 1 point |

Given this block of code

Text

Description automatically generated

Which of the assertions below is true?

|  |  |
| --- | --- |
|  | The method numOfKids() in class kindergarten must have a body. |
|  | You must add a return statement in method numOfKids(). |
|  | keywords abstract and public cannot be used together. |
|  | Class kindergarten must be defined as abstract. |
| **Question 40** | |  | 1 / 1 point |

Based on this code:

Text

Description automatically generated

which of the two code fragments shown below will compile?

1. interface High implements Low{}  
2. abstract class MidPoint extends Low  
    { public boolean m1(){ return true;}  
3. abstract class MidPoint implements Low{}  
4. abstract class MidPoint implements Low  
    {public boolean m1(){ return (8 > 5);}  
5. abstract class MidPoint implements Low  
    {protected boolean m1() { return (3 < 5);}

|  |  |
| --- | --- |
|  | 4 and 5 |
|  | 3 and 4 |
|  | 1 and 3 |
|  | 1 and 2 |
|  | 2 and 3 |
| **Question 41** | |  | 0 / 1 point |

Declaring a class abstract in Java is useful:

|  |  |
| --- | --- |
|  | when it is not desirable to have default implementations of some of the methods |
|  | to make programmers to extend that class to use its capabilities |
| Incorrect Response | When it is meaningful to have objects of that class |
|  | To keep programmers from extending the class. |
| Correct Answer | In situations where it is meaningless to create objects of that class. |
| **Question 42** | |  | 1 / 1 point |

Which of these declares an abstract method in an existing abstract Java class?

|  |  |
| --- | --- |
|  | public abstract myMethod(); |
|  | public abstract void myMethod(); |
|  | public void abstract myMethod(); |
|  | public abstract void  myMethod() { }; |
|  | public void myMethod() { }; |
| **Question 43** | |  | 1 / 1 point |

Choose the class definition that defines a legal abstract class.

|  |  |
| --- | --- |
|  | class D { abstract void incomplete(); } |
|  | abstract class D { abstract void incomplete(); } |
|  | public class abstract D { abstract void incomplete(); } |
|  | class D { abstract void incomplete() { }; } |
| **Question 44** | |  | 1 / 1 point |

What element is at position multiArr [2][3]?  
int [ ] [ ] multiArr =  {  {2, 3, 4, 5},  {6, 7, 8, 9},  {10, 11, 12, 13},  {14, 15, 16, 17}  };

|  |  |
| --- | --- |
|  | 13 |
|  | 8 |
|  | 12 |
|  | 16 |
| **Question 45** | |  | 1 / 1 point |

What is does the program below output?

Text

Description automatically generated

|  |  |
| --- | --- |
|  | a runtime error caused by e [1] in the output statement |
|  | a runtime error caused by failing to initialize array elements |
|  | a compile time error since an incorrect array size was specified in the array declaration |
|  | The program compiles and runs fine and displaying e[0] is 0.0 as output |
| **Question 46** | |  | 0 / 1 point |

What would the following program code output?

Text

Description automatically generated

|  |  |
| --- | --- |
| Correct Answer | 1 |
|  | 3 |
| Incorrect Response | 6 |
|  | 2 |
| **Question 47** | |  | 4 / 4 points |

Match "Subclass Same Package" members to it's visibility

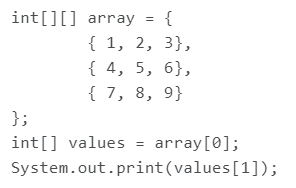
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  | \_\_1\_\_ |  | public | |  | \_\_1\_\_ |  | default (no modifier) | |  | \_\_2\_\_ |  | private | |  | \_\_1\_\_ |  | Protected | |

|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 4 / 4 points |

Match "Subclass Different Package" member to it's visibility

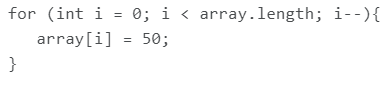
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  | \_\_2\_\_ |  | private | |  | \_\_1\_\_ |  | protected | |  | \_\_2\_\_ |  | default (no modifier) | |  | \_\_1\_\_ |  | public | |  | |  |  | | --- | --- | | **1**. | Visible | | **2**. | Not visible | | | |
| **Question 2** | | |  | 1 / 1 point |

What is result of the following Java code?



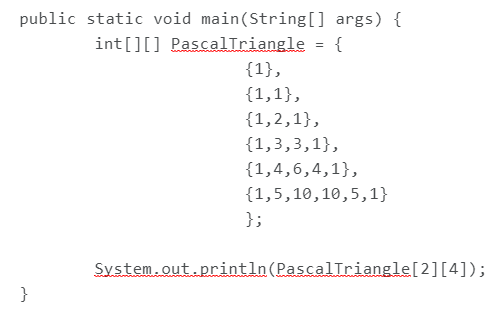
|  |  |  |  |
| --- | --- | --- | --- |
|  | 5 | | |
|  | ArrayIndexOutOfBoundsException | | |
|  | 2 | | |
|  | 4 | | |
|  | compiler error | | |
| **Question 3** | |  | 1 / 1 point | |

You have an array named array of type int and of size 10. What is true of the following Java code?



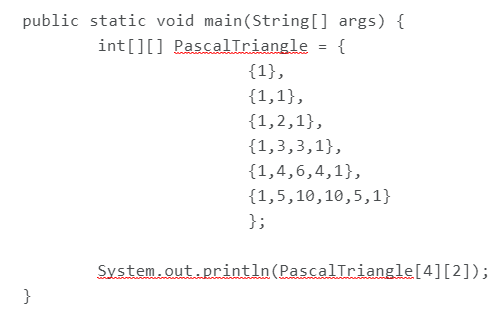
|  |  |  |  |
| --- | --- | --- | --- |
|  | it would result in an IndexOutOfBoundsException being thrown | | |
|  | it would result in a compile-time error | | |
|  | it would assign all of the elements of array the value 0 | | |
|  | it would assign all of the elements of array the value 50 | | |
| **Question 4** | |  | 1 / 1 point | |

What is the output of the following?:



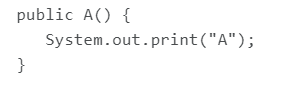
|  |  |  |  |
| --- | --- | --- | --- |
|  | runtime exception | | |
|  | 6 | | |
|  | 4 | | |
|  | nothing, the code won’t compile | | |
|  | 3 | | |
| **Question 5** | |  | 1 / 1 point | |

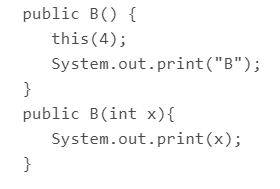
What is the output of the following?:



|  |  |  |  |
| --- | --- | --- | --- |
|  | 6 | | |
|  | nothing, the code won’t compile | | |
|  | 3 | | |
|  | runtime exception | | |
|  | 4 | | |
| **Question 6** | |  | 1 / 1 point | |

You have a class named A with the following in it:



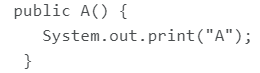
You have a class named B that extends class A, and has the following in it:  


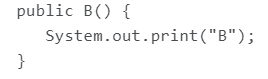
What is printed out by the following Java statement:

B b = new B();

|  |  |  |  |
| --- | --- | --- | --- |
|  | None of these | | |
|  | AB4 | | |
|  | AB | | |
|  | 4AB | | |
|  | A4B | | |
| **Question 7** | |  | 1 / 1 point | |

You have a class named A with the following in it:



You have a class named B that extends class A, and has the following in it:  


What is printed out by the following Java statement:

A a = new B();

|  |  |  |  |
| --- | --- | --- | --- |
|  | B | | |
|  | BA | | |
|  | AB | | |
|  | A | | |
|  | None of these | | |
| **Question 8** | |  | 1 / 1 point | |

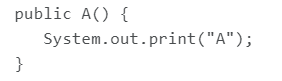
Can a constructor call another constructor of the same class?

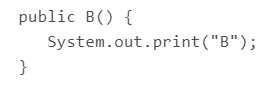
|  |  |  |  |
| --- | --- | --- | --- |
|  | Yes, and it can be anywhere in the constructor | | |
|  | Yes, but the call must be the last statement of the constructor | | |
|  | Never | | |
|  | Yes, but the call must be the first statement of the constructor | | |
| **Question 9** | |  | 1 / 1 point | |

The default constructor of a subclass always calls the default constructor of it's superclass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | True | | |
|  |  | False | | |
| **Question 10** | | |  | 1 / 1 point | |

You have a class named A with the following in it:



You have a class named B that extends class A, and has the following in it:  


Which of the following statements would result in a runtime error?

|  |  |  |  |
| --- | --- | --- | --- |
|  | A a = new B(); | | |
|  | A a = (A) new B(); | | |
|  | A a = new A(); | | |
|  | A a = (B) new A(); | | |
| **Question 11** | |  | 1 / 1 point | |

The root interface of Java collection framework hierarchy is:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Root | | |
|  | List | | |
|  | Collection | | |
|  | Set | | |
|  | Collections | | |
| **Question 12** | |  | 0 / 1 point | |

Which of these helps ***efficiently*** insert elements at a specific position in a collection?

|  |  |  |  |
| --- | --- | --- | --- |
| Incorrect Response | List | | |
| Correct Answer | LinkedList | | |
|  | ArrayList | | |
|  | Vector | | |
| **Question 13** | |  | 1 / 1 point | |

Methods such as reverse, shuffle are offered in:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Object | | |
|  | Collection | | |
|  | Collections | | |
|  | All of these | | |
| **Question 14** | |  | 1 / 1 point | |

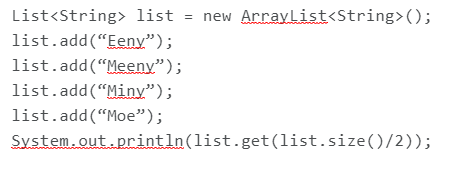
Which of those allows duplicate elements?

|  |  |  |  |
| --- | --- | --- | --- |
|  | List | | |
|  | Collections | | |
|  | Set | | |
|  | Collection | | |
| **Question 15** | |  | 0 / 1 point | |

What should we use when add and remove operations are more frequent than get operations

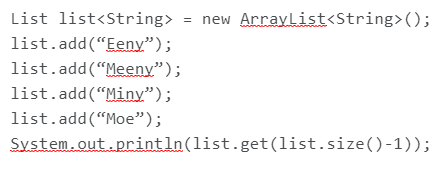
|  |  |  |  |
| --- | --- | --- | --- |
| Correct Answer | LinkedList | | |
| Incorrect Response | They are all the same | | |
|  | Vector | | |
|  | ArrayList | | |
| **Question 16** | |  | 1 / 1 point | |

What is the output of the following code segment:



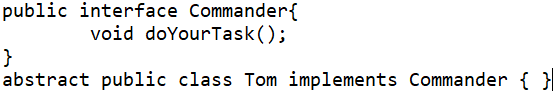
|  |  |  |  |
| --- | --- | --- | --- |
|  | Eeny | | |
|  | Meeny | | |
|  | Miny | | |
|  | Moe | | |
|  | Throw an exception | | |
| **Question 17** | |  | 1 / 1 point | |

What is the output of the following code segment:



|  |  |  |  |
| --- | --- | --- | --- |
|  | Eeny | | |
|  | Meeny | | |
|  | Miny | | |
|  | Moe | | |
|  | Throw an exception | | |
| **Question 18** | |  | 1 / 1 point | |

Given the following piece of code:



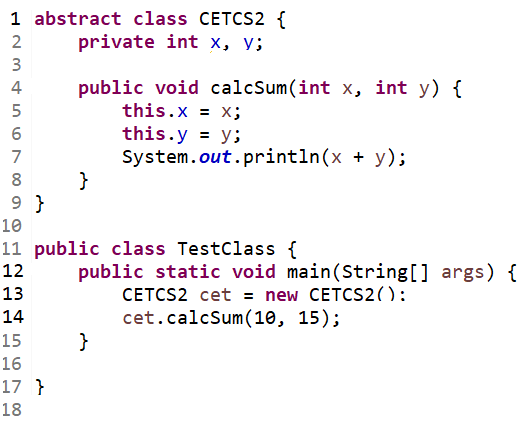
which of the the statements below is correct?

|  |  |  |  |
| --- | --- | --- | --- |
|  | The code won't compile since class Tom must implement method doYourTask() from interface Commander. | | |
|  | The program will not compile since declaration of class Tom uses keyword implements instead of extends. | | |
|  | The program will compile without errors. | | |
|  | The program will not compile since method doYourTask() in interface Commander must be defined as abstract. | | |
| **Question 19** | |  | 0 / 1 point | |

Which of these declares an abstract method in an existing abstract Java class?

|  |  |  |  |
| --- | --- | --- | --- |
|  | public abstract myMethod(); | | |
| Incorrect Response | public abstract void  myMethod() { }; | | |
|  | public void abstract myMethod(); | | |
| Correct Answer | public abstract void myMethod(); | | |
|  | public void myMethod() { }; | | |
| **Question 20** | |  | 0 / 1 point | |

What will happen when this program is compiled?



|  |  |  |  |
| --- | --- | --- | --- |
|  | None of these | | |
|  | runtime error due line 2 | | |
|  | compilation error due to line 7 | | |
| Correct Answer | compilation error due to line 13 | | |
| Incorrect Response | run successfully and print 25 | | |
| **Question 21** | |  | 0 / 1 point | |

Select the statement which is true about abstract classes

|  |  |  |  |
| --- | --- | --- | --- |
| Correct Answer | a subclass of a non abstract superclass can be abstract | | |
|  | an abstract class cannot be extended | | |
| Incorrect Response | an abstract class can be instantiated | | |
|  | all the statements given are true | | |
| **Question 22** | |  | 1 / 1 point | |

Suppose B is an abstract class, C is a concrete subclass of B, and both B and C have a no-arg constructor. Which of the following is correct?

1. B  b = new B();

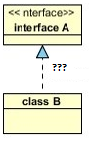
2. B  b = new C ();

3. C  c = new B();

4. C  c = new C();

|  |  |  |  |
| --- | --- | --- | --- |
|  | 2 and 4 | | |
|  | 2 and 3 | | |
|  | 1 and 3 | | |
|  | 3 and 4 | | |
| **Question 23** | |  | 1 / 1 point | |

What is the best word for the relationship?

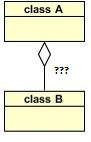


|  |  |  |  |
| --- | --- | --- | --- |
|  | uses | | |
|  | owns | | |
|  | realizes | | |
|  | is-a | | |
|  | contains | | |
| **Question 24** | |  | 1 / 1 point | |

Which symbol is generalization ?

|  |  |  |  |
| --- | --- | --- | --- |
|  | Arrow with dotted line | | |
|  | None-triangular Arrow | | |
|  | Empty diamond | | |
|  | Filled in diamond | | |
|  | Triagular Arrow | | |
| **Question 25** | |  | 1 / 1 point | |

What is the best word for the relationship?



|  |  |  |  |
| --- | --- | --- | --- |
|  | contains | | |
|  | uses | | |
|  | owns | | |
|  | is-a | | |
|  | realizes | | |
| **Question 26** | |  | 1 / 1 point | |

Which symbol is aggregation?

|  |  |  |  |
| --- | --- | --- | --- |
|  | Filled in diamond | | |
|  | None-triangular Arrow | | |
|  | Empty diamond | | |
|  | Triagular Arrow | | |
|  | Arrow with dotted line | | |
| **Question 27** | |  | 1 / 1 point | |

The statement that opens file "trans.ser" for input using ObjectInputStream variable inTrans to wrap a FileInputStream object is:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | ObjectInputStream inTrans = ObjectInputStream ("trans.ser"); | | | |
|  | |  |  | | --- | --- | |  | ObjectInputStream inTrans = new ObjectInputStream ("trans.ser"); | | | |
|  | |  |  | | --- | --- | |  | ObjectInputStream inTrans = new ObjectInputStream (new FileInputStream("trans.ser")); | | | |
|  | |  |  | | --- | --- | |  | none of the above | | | |
| **Question 28** | |  | 1 / 1 point | |

Which of the following classes is *not* used for file input?

|  |  |  |  |
| --- | --- | --- | --- |
|  | Formatter | | |
|  | ObjectInputStream | | |
|  | FileInputStream | | |
|  | FileReader | | |
| **Question 29** | |  | 1 / 1 point | |

The statement(s) that reads a record from the file "trans.ser" into an object of class TransactionRecord (using the ObjectInputStream variable inTrans which has been declared successfully already) is:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | TransactionRecord trans = inTrans.readObject(); | | | |
|  | |  |  | | --- | --- | |  | TransactionRecord trans = (TransactionRecord) inTrans.readObject(); | | | |
|  | |  |  | | --- | --- | |  | TransactionRecord trans = (TransactionRecord) inTrans.writeObject(); | | | |
|  | |  |  | | --- | --- | |  | none of the above. | | | |
| **Question 30** | |  | 1 / 1 point | |

The statement that opens file "newTrans.ser" for output using ObjectOutputStream variable outTrans to wrap a FileOutputStream object is:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | ObjectOutputStream outTrans = new ObjectOutputStream (new FileOutputStream ("newtrans.ser")); | | | |
|  | |  |  | | --- | --- | |  | ObjectOutputStream outTrans = new ObjectInputStream (new FileInputStream ("newtrans.ser")); | | | |
|  | |  |  | | --- | --- | |  | ObjectInputStream outTrans = new ObjectOutputStream (new FileOutputStream ("newtrans.ser")); | | | |
|  | |  |  | | --- | --- | |  | none of the above | | | |
| **Question 31** | |  | 1 / 1 point | |

If an absolute path of a file is NOT specified when declaring a file object and connecting to that file to read from it, the file must exist be in the project directory.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 1) True | | |
|  |  | 2) False | | |
| **Question 32** | | |  | 1 / 1 point | |

Which of the following is **false**:

|  |  |  |  |
| --- | --- | --- | --- |
|  | You can extend a concrete class to an abstract class | | |
|  | To use you need to extend an abstract class to a concrete class. | | |
|  | An abstract class can have any constructors | | |
|  | An abstract class must have at least one abstract method. | | |
| **Question 33** | |  | 0 / 1 point | |

Which of the following statements about Abstract Classes and Interfaces are true?

|  |  |  |  |
| --- | --- | --- | --- |
|  | an abstract class doesn’t include implementation (method bodies) for any of its method definitions | | |
| Incorrect Response | all of these | | |
| Correct Answer | all attributes in an interface, if any, must be constants (i.e. public static final) | | |
|  | a class can extend more than one abstract class, but a class can implement only one interface | | |
| **Question 34** | |  | 1 / 1 point | |

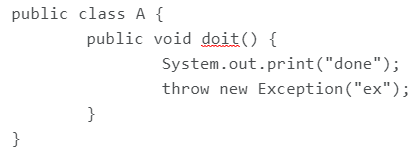
When we are defining a new interface, what happens if we leave out the abstract keyword in the method definitions?

|  |  |  |  |
| --- | --- | --- | --- |
|  | Because the abstract keyword is for abstract classes, a compilation error would result. | | |
|  | A compilation error would result because abstract methods are not allowed in an interface. | | |
|  | Because all methods in an interface are abstract, it doesn’t matter whether or not we specify the abstract keyword. | | |
|  | Because all methods in an interface are abstract, it would cause a compilation error. | | |
| **Question 35** | |  | 1 / 1 point | |

What Java keyword is used to inherit an interface?

|  |  |  |  |
| --- | --- | --- | --- |
|  | extends | | |
|  | overrides | | |
|  | overloads | | |
|  | implements | | |
|  | inherits | | |
| **Question 36** | |  | 1 / 1 point | |

Given this class definition



what would be printed by the following Java statements?

A a = new A();

a.doit();

|  |  |  |  |
| --- | --- | --- | --- |
|  | None of these answers | | |
|  | ex | | |
|  | nothing, the code won’t compile. | | |
|  | done | | |
|  | doneex | | |
| **Question 37** | |  | 0 / 1 point | |

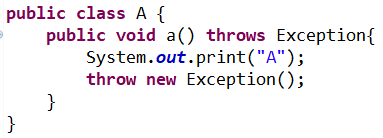
Which of the following statement is most true:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | | a) | The **throws** keyword is used in the signature of a method to indicate that this method might throw an exception | | | |
| Incorrect Response | |  |  | | --- | --- | | b) | The **throw** keyword is used to explicitly throw an exception from a method | | | |
|  | |  |  | | --- | --- | | c) | All exceptions that are not “Error” **must** be handled | | | |
|  | |  |  | | --- | --- | | d) | Catch blocks **must** be placed from the general to the specific | | | |
| Correct Answer | |  |  | | --- | --- | | e) | a) and b) above | | | |
| **Question 38** | |  | 0 / 1 point | |

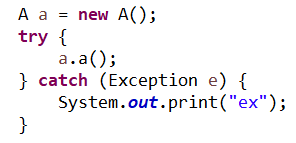
Which of the following statement is most true:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | | a) | The **throwable** keyword is used in the signature of a method to indicate that this method might throw an exception | | | |
|  | |  |  | | --- | --- | | b) | The **throwException** keyword is used to explicitly throw an exception from a method | | | |
| Correct Answer | |  |  | | --- | --- | | c) | RuntimeException and its subclasses are **unchecked** exceptions | | | |
|  | |  |  | | --- | --- | | d) | Catch blocks **must** be placed from the general to the specific | | | |
| Incorrect Response | |  |  | | --- | --- | | e) | a) and b) above | | | |
| **Question 39** | |  | 1 / 1 point | |

Given this class definition

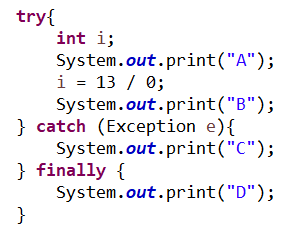


what would be printed out by the following Java Statements?



|  |  |  |  |
| --- | --- | --- | --- |
|  | nothing, the code won’t compile. | | |
|  | A | | |
|  | Aex | | |
|  | exA | | |
|  | None of these answers | | |
| **Question 40** | |  | 1 / 1 point | |

In the following Java code fragment



what is printed out, assuming the division by zero generates an exception?

|  |  |  |  |
| --- | --- | --- | --- |
|  | ACD | | |
|  | CD | | |
|  | AD | | |
|  | ADCB | | |
|  | AC | | |
| **Question 41** | |  | 1 / 1 point | |

What is true of class variables in Java that are declared to be static?

|  |  |  |  |
| --- | --- | --- | --- |
|  | They are associated with the class in which they are declared as opposed to objects that are instances of the class | | |
|  | They are a property of every object that is an instance of the class | | |
|  | They are immutable and cannot be changed | | |
|  | They are implemented on the stack | | |
| **Question 42** | |  | 1 / 1 point | |

What is true of class variables in Java that are declared to be static?

|  |  |  |  |
| --- | --- | --- | --- |
|  | They are implemented on the stack | | |
|  | There is only one copy of a static class variable, shared by all instances | | |
|  | They are a property of every object that is an instance of the class | | |
|  | They are immutable and cannot be changed | | |
| **Question 43** | |  | 1 / 1 point | |

When working with a Scanner in Java, its next method will return

|  |  |  |  |
| --- | --- | --- | --- |
|  | None of these answers | | |
|  | The next word of input | | |
|  | The next Unicode character | | |
|  | The next integer value | | |
|  | The next line of input | | |
| **Question 44** | |  | 1 / 1 point | |

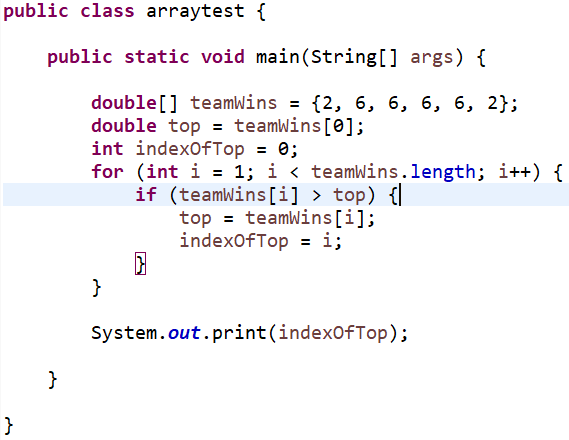
Which of the following statements is **true**:

|  |  |  |  |
| --- | --- | --- | --- |
|  | A static variable cannot be changed | | |
|  | A multidimensional array can have more than three dimensions | | |
|  | Polymorphism refers to the bundling of data with the methods that operate on that data | | |
|  | A List can contain both primitives and reference types | | |
|  | A package is intended to group related objects | | |
| **Question 45** | |  | 1 / 1 point | |

What element is at position multiArr [2][3]?  
int [ ] [ ] multiArr =  {  {2, 3, 4, 5},  {6, 7, 8, 9},  {10, 11, 12, 13},  {14, 15, 16, 17}  };

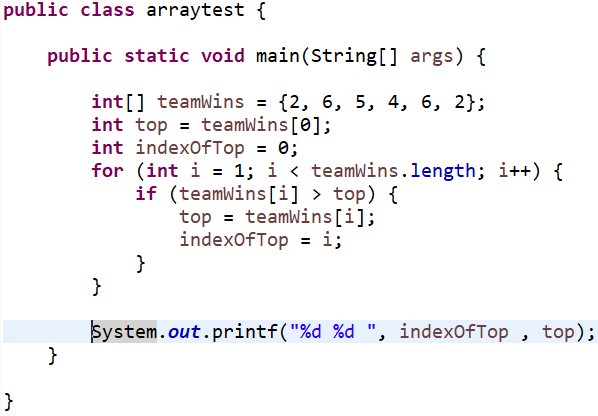
|  |  |  |  |
| --- | --- | --- | --- |
|  | 13 | | |
|  | 16 | | |
|  | 8 | | |
|  | 12 | | |
| **Question 46** | |  | 0 / 1 point | |

What would the following program code output?



|  |  |  |  |
| --- | --- | --- | --- |
|  | 3 | | |
|  | 2 | | |
| Incorrect Response | 6 | | |
| Correct Answer | 1 | | |
| **Question 47** | |  | 1 / 1 point | |

What is the output of the program shown below?



|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 6 | | |
|  | 1 5 | | |
|  | 1 4 | | |
|  | 1 2 | | |
| **Question 1** | |  | 1 / 1 point |

In MVC (Model-View-Controller) pattern, which block of application deals with main data/application.

|  |  |  |  |
| --- | --- | --- | --- |
|  | view | | |
|  | model | | |
|  | controller | | |
| **Question 2** | |  | 1 / 1 point | |

Which category of design patterns characterize the ways in which classes or objects  interact and distribute responsibility.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Structural | | |
|  | Behavioural | | |
|  | Creational | | |
| **Question 3** | |  | 1 / 1 point | |

Which design pattern ensures that there exists only one instance of a class within the system.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Factory | | |
|  | Singleton | | |
|  | Decorator | | |
|  | MVC | | |
| **Question 4** | |  | 1 / 1 point | |

Which design pattern is also known as Wrapper with the intention to attach additional responsibilities to an object dynamically.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Singleton | | |
|  | Decorator | | |
|  | Factory | | |
|  | Facade | | |
| **Question 5** | |  | 1 / 1 point | |

In MVC (Model-View-Controller) pattern, which block of application deals with the presentation of information.

|  |  |  |  |
| --- | --- | --- | --- |
|  | model | | |
|  | controller | | |
|  | view | | |
| **Question 6** | |  | 1 / 1 point | |

In MVC (Model-View-Controller) pattern, which block of application deals with

controlling the logic and behavior of the application.

|  |  |  |  |
| --- | --- | --- | --- |
|  | view | | |
|  | model | | |
|  | controller | | |
| **Question 7** | |  | 1 / 1 point | |

Which design pattern provides a way to access the elements of an aggregate object sequentially without  exposing its underlying representation.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Iterator | | |
|  | Singleton | | |
|  | Decorator | | |
|  | Factory | | |
| **Question 8** | |  | 1 / 1 point | |

Which pattern returns an instance of one of several possible classes depending on the  data provided to it.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | Decorator | | |
|  | | | Singleton | | |
|  | | | Factory | | |
|  | | | Iterator | | |
|  |  |
|  | |
| **Question 1** | | | |  | 0 / 1 point |

JUnit is primarily used for:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Acceptance testing | | |
| Correct Answer | Unit testing | | |
|  | Integration testing | | |
| Incorrect Response | All of the others | | |
|  | System testing | | |
| **Question 2** | |  | 1 / 1 point | |

Which of the following is correct about JUnit?

|  |  |  |  |
| --- | --- | --- | --- |
|  | JUnit tests can be organized into test suites containing test cases and even other test suites. | | |
|  | JUnit shows test progress in a bar that is green if test is going fine and it turns red when a test fails in eclipse. | | |
|  | Both of the above | | |
|  | None of the above. | | |
| **Question 3** | |  | 1 / 1 point | |

To sort a Java ArrayList, you would use the sort method of which of the following?

|  |  |  |  |
| --- | --- | --- | --- |
|  | The Collections class | | |
|  | The Collection interface | | |
|  | The Arrays class | | |
|  | The List interface | | |
|  | The ArrayList class | | |
| **Question 4** | |  | 1 / 1 point | |

java.util.Collection is a:

|  |  |  |  |
| --- | --- | --- | --- |
|  | None of these | | |
|  | Interface | | |
|  | Class | | |
|  | Object | | |
| **Question 5** | |  | 1 / 1 point | |

Which of these helps ***efficiently*** remove a element at a specific position in a collection?

|  |  |  |  |
| --- | --- | --- | --- |
|  | LinkedList | | |
|  | Vector | | |
|  | ArrayList | | |
|  | List | | |
| **Question 6** | |  | 1 / 1 point | |

Which of the following is an “unchecked exception”?

|  |  |  |  |
| --- | --- | --- | --- |
|  | RuntimeException | | |
|  | Error | | |
|  | IOException | | |
|  | All of the above | | |
|  | None of the above | | |
| **Question 7** | |  | 1 / 1 point | |

When should a program *explicitly* use the **this** reference?

|  |  |  |  |
| --- | --- | --- | --- |
|  | Accessing a private variable. | | |
|  | Accessing a field that has the same name as a local variable. | | |
|  | Accessing a public variable | | |
|  | Accessing a local variable | | |
| **Question 8** | |  | 0 / 1 point | |

What is true of class variables in Java that are declared to be static?

|  |  |  |  |
| --- | --- | --- | --- |
|  | They are implemented on the stack | | |
| Incorrect Response | They are a property of every object that is an instance of the class | | |
| Correct Answer | There is only one copy of a static class variable, shared by all instances | | |
|  | They are immutable and cannot be changed | | |
| **Question 9** | |  | 1 / 1 point | |

Overriding a method differs from overloading a method because:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Overloaded methods have the same signature. | | |
|  | Overridden methods have the same signature. | | |
|  | Both of the above. | | |
|  | Neither of the above. | | |
| **Question 10** | |  | 1 / 1 point | |

When working with a Scanner in Java, its next method will return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | The next integer value | | |
|  | | | The next Unicode character | | |
|  | | | None of these answers | | |
|  | | | The next word of input | | |
|  | | | The next line of input | | |
|  |  |
|  | |
| **Question 1** | | | |  |  |

 Which of the following describes the Iterator pattern correctly?

|  |  |
| --- | --- |
|  | This pattern is used to reduce communication complexity between multiple objects or classes. |
|  | This pattern is used when there is one-to-many relationship between objects such as if one object is modified, its depenedent objects are to be notified automatically. |
|  | This pattern is used to get a way to access the elements of a collection object in sequential manner without any need to know its underlying representation |
|  | This pattern is used to restore state of an object to a previous state. |
| **Question 2** | | |  |  |

Which design pattern ensures that only one object of particular class gets created?

|  |  |
| --- | --- |
|  | Adapter |
|  | Decorator |
|  | Factory Method |
|  | Singleton |
| **Question 3** | | |  |  |

Which of the pattern attach additional responsibilities to an object dynamically

|  |  |
| --- | --- |
|  | Factory Method |
|  | Singleton |
|  | Decorator |
|  | Iterator |
| **Question 4** | | |  |  |

MVC pattern stands for?

|  |  |
| --- | --- |
|  | Model View Controller |
|  | Model View Control |
|  | Model View Class |
|  | Mock View Class |
| **Question 5** | | |  |  |

In MVC pattern, which one of the following represents the visualization of the data that the model contains

|  |  |
| --- | --- |
|  | View |
|  | Model |
|  | Controller |
|  | All of these |
| **Question 6** | | |  |  |

Which of the below is not a valid classification of design pattern?

|  |  |
| --- | --- |
|  | behavioural pattern |
|  | Structural pattern |
|  | Colelction pattern |
|  | Creational pattern |
| **Question 7** | | |  |  |

Which type of design patterns concern class and object composition?

|  |  |
| --- | --- |
|  | Behavioral Design Pattern |
|  | Structural Design Patterns |
|  | Creational Design Patterns |
| **Question 8** | | |  |  |

Which of the below is not a valid design pattern?

|  |  |
| --- | --- |
|  | Command |
|  | Factory |
|  | Java |
|  | Iterator |
| **Question 9** | | |  |  |

Which of the below author is not a part of GOF (Gang of Four)?

|  |  |
| --- | --- |
|  | Ralph Johnson |
|  | Erich Gamma |
|  | John William |
|  | Richard Helm |
| **Question 10** | | |  |  |

Which of the following describes the Factory pattern correctly?

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | This pattern involves a single class which is responsible to create an object while making sure that only single object gets created. |
|  | | | This pattern creates object without exposing the creation logic to the client and refer to newly created object using a common interface. |
|  | | | In this pattern an interface is responsible for creating a factory of related objects without explicitly specifying their classes. |
|  | | | This pattern is used when we want to pass data with multiple attributes in one shot from client to server. |
|  |  |