1. Given:

1. public class **Test** {

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

3. class **Foo** {

4. public int i = 3;

5. }

6. **Object** o = (**Object**)new **Foo**();

7. Foo foo = (Foo)o;

8. System.out.**println**(“i = “ + foo.i);

9. }

10. }

What is the result?

* 1. **i = 3**
  2. Compilation fails
  3. ClassCastException thrown at line6
  4. ClassCastException thrown at line7

1. Which two cause a compiler error? (Choose two)
   1. float[] = new float(3);
   2. float f2[] = new float[];
   3. float[] f1 = new float[3];
   4. float f3[] = new float[3];
   5. float f5[] = { 1.0f, 2.0f, 2.0f };
   6. float f4[] = new float[] { 1.0f. 2.0f. 3.0f};
   7. **A,B**
   8. B,C
   9. C,D
   10. D,E
2. Given:  
   1. class Test {   
   2. private Demo d;   
   3. void start() {   
   4. d = new Demo();   
   5. this.takeDemo(d);   
   6. }   
   7.  
   8. void takeDemo(Demo demo) {   
   9. demo = null;   
   10. demo = new Demo();   
   11. }   
   12. }   
     
   When is the Demo object, created on line 3, eligible for garbage collection?
   1. After line 5.
   2. After line 9.
   3. After the start() method completes.
   4. When the takeDemo() method completes.
   5. **When the instance running this code is made eligible for garbage collection.**
3. Which three form part of correct array declarations? (Choose three)
   1. **public int a []**
   2. **static int [] a**
   3. public [] int a
   4. private int a [3]
   5. **public final int [] a**
4. Given:

1. class Base {

2. Base() { System.out.print(“Base”); }

3. }

4. public class Alpha extends Base {

5. public static void main( String[] args ) {

6. new Alpha();

7. new Base();

8. }

9. }

What is the result?

* 1. Base
  2. **BaseBase**
  3. Compilation fails.
  4. The code runs with no output.
  5. An exception is thrown at runtime.

1. Given:

1. public class Test {

2. public static void main(String[] args) {

3. int x = 0;

4. assert (x > 0) ? “assertion failed” : “assertion passed”;

5. System.out.println(“Finished”);

6. }

7. }

What is the result?

* 1. finished
  2. **Compilation fails.**
  3. An AssertionError is thrown and finished is output.
  4. An AssertionError is thrown with the message “assertion failed”.
  5. An AssertionError is thrown with the message “assertion passed”.

1. Which statement is true?
   1. Programs will not run out of memory.
   2. Objects that will never again be used are eligible for garbage collection.
   3. Objects that are referred to by other objects will never be garbage collected.
   4. **Objects that can be reached from a live thread will never be garbage collected.**
   5. An AssertionError is thrown with the message “assertion passed”.
2. Given:

1. void start() {

2. A a = new A();

3. B b = new B();

4. a.s(b);

5. b = null;

6. a = null;

7. System.out.println(“start completed”);

8. }

When is the B object, created in line 14, eligible for garbage collection?

* 1. After line 16.
  2. AfterLine 17
  3. After line 18 (when method ends)
  4. **There is no way to be absolutely certain**
  5. The object is not eligible for Garbage Colletion

1. Given:

1. package cts;

2. public class Outer {

3. public static class Inner {

4. }

5. }

Which statement is true?

* 1. Compilation fails.
  2. An instance of the Inner class can be constructed with “new Outer.Inner()”.
  3. **An instance of the Inner class cannot be constructed outside of package foo.**
  4. An instance of the Inner class can be constructed only from within the Outer class.
  5. From within the package foo, and instance of the Inner class can be constructed with “new Inner()”.

1. When Overloading does not occur?
   1. More than one method with same name but different method signature and different number or type of parameters
   2. More than one method with same name, same signature but different number of signature
   3. More than one method with same name, same signature, same number of parameters but different type
   4. **More than one method with same name, same number of parameters and type but different signature**
2. Which concept of Java is achieved by combining methods and attribute into a class?
   1. **Encapsulation**
   2. Inheritance
   3. Polymorphism
   4. Abstraction
3. What is it called where child object gets killed if parent object is killed?
   1. Aggregation
   2. **Composition**
   3. Encapsulation
   4. Association
4. class main\_class

{

public static void main(String args[])

{

int x = 9;

if (x == 9)

{

int x = 8;

System.out.println(x);

}

}

}

* 1. 9
  2. 8
  3. **Compilation Error**
  4. Runtime error

1. Given:

1. public class Foo {

2. public static void main (String [] args) {

3. StringBuffer a = new StringBuffer (“A”);

4. StringBuffer b = new StringBuffer (“B”);

5. operate (a,b);

6. system.out.printIn{a + “,” +b};

7. )

8. static void operate (StringBuffer x, StringBuffer y) {

9. x.append {y};

10. y = x;

11. )

12. }

What is the result?

* 1. The code compiles and prints “A,B”.
  2. The code compiles and prints “A,A”.
  3. The code compiles and prints “B,B”.
  4. **The code compiles and prints “AB,B”.**
  5. The code compiles and prints “AB,AB”.

1. exhibit:

1. public class test {

2. public static void add3 (Integer i) }

3. int val = i.intValue ( );

4. val += 3;

5. i = new Integer (val);

6. }

7.

8. public static void main (String args [ ] ) {

9. Integer i = new Integer (0);

10. add3 (i);

11. system.out.printIn (i.intValue ( ) );

12. }

13. )

What is the result?

* 1. Compilation will fail
  2. **The program prints "0"**
  3. The program prints "3"
  4. Compilation will succeed but an exception will be thrown at line 3.

1. How to sort elements of ArrayList?
   1. Collection.sort(listObj);
   2. **Collections.sort(listObj);**
   3. listObj.sort();
   4. Sorter.sortAsc(listObj);
2. How to remove duplicates from List?
   1. **HashSet<String> listToSet = new HashSet<String>(duplicateList);**
   2. HashSet<String> listToSet = duplicateList.toSet();
   3. HashSet<String> listToSet = Collections.convertToSet(duplicateList);
   4. HashSet<String> listToSet = duplicateList.getSet();
3. 1. interface foo {

2. int k = 0;

3. ]

4.

5. public class test implements Foo (

6. public static void main(String args[]) (

7. int i;

8. Test test = new test ();

9. i= test.k;

10.i= Test.k; 11.i= Foo.k; 12.)

13.)

What is the result?

* 1. **Compilation succeeds.**
  2. An error at line 2 causes compilation to fail.
  3. An error at line 9 causes compilation to fail.
  4. An error at line 10 causes compilation to fail.

1. What is the prototype of the default constructor of this class?

public class prototype { }

* 1. prototype( )
  2. prototype(void)
  3. public prototype(void)
  4. **public prototype( )**

Given:

8. int index = 1;

9. boolean[] test = new Boolean[3];

10. boolean foo= test [index];

What is the result?

* 1. Foo has the value of 0.
  2. Foo has the value of null.
  3. Foo has the value of true.
  4. **Foo has the value of false.**

1. Which will declare a method that forces a subclass to implement it?
   1. public double methoda();
   2. static void methoda (double d1) {} Public native double methoda();
   3. protected void methoda (double d1){}
   4. **abstract public void methoda();**
2. You want subclasses in any package to have access to members of a superclass. Which is the most restrictive access modifier that will accomplish this objective?
   1. Public
   2. Private
   3. **Protected**
   4. Trainscient
3. Given:

2. abstract class abstrctIt {

3. abstract float getFloat ();

4. )

5. public class AbstractTest extends AbstractIt {

6. private float f1= 1.0f;

7. private float getFloat () {return f1;}

8. }

What is the result?

* 1. Compilation is successful.
  2. An error on line 6 causes a runtime failure.
  3. **An error at line 6 causes compilation to fail.**
  4. An error at line 2 causes compilation to fail.

1. What is the error in this code?

byte b = 50;

b = b \* 50;

* 1. b can not contain value 100, limited by its range.
  2. **\* operator has converted b \* 50 into int, which can not be converted to byte without casting.**
  3. b can not contain value 50.
  4. No error in this code

1. Exhibit:

2. import java.io.IOException;

3. public class ExceptionTest(

4. public static void main (String[]args)

5. try (

6. methodA();

7. ) catch (IOException e) (

8. system.out.printIn(“Caught IOException”);

9. ) catch (Exception e) (

10. system.out.printIn(“Caught Exception”);

11. )

12. )

13. public void methodA () {

14. throw new IOException ();

15. )

16. )

What is the result?

* 1. **The code will not compile.**
  2. The output is caught exception.
  3. The output is caught IOException
  4. The program executes normally without printing a message.

1. If an expression contains double, int, float, long, then whole expression will promoted into which of these data types?
   1. long
   2. int
   3. **double**
   4. float
2. What is the output of this program?

class A

   {

       final public int calculate(int a, int b) { return 1; }

   }

   class B extends A

   {

       public int calculate(int a, int b) { return 2; }

   }

    public class output

    {

       public static void main(String args[])

       {

           B object = new B();

           System.out.print("b is " + b.calculate(0, 1));

       }

   }

* 1. b is : 2
  2. b is : 1
  3. **Compilation Error**
  4. Runtime error

1. Which component is used to compile, debug and execute java program?
   1. JVM
   2. **JDK**
   3. JIT
   4. JRE
2. Given:

1 package test1;

2 public class Test1 {

3 static int x = 42;

4 }

1 package test2;

2 public class Test2 extends test1.Test1 {

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

4 System.out.println("x = " + x);

5 }

6 }

What is the result?

* 1. X=0
  2. X=42
  3. **Compilation fails because of an error in line 2 of class Test2.**
  4. Compilation fails because of an error in line 3 of class Test1.

1. Given:

1 public class Test {

2 public int aMethod() {

3 static int i = 0;

4 i++;

5 return i;

6 }

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

8 Test test = new Test();

9 test.aMethod();

10 int j = test.aMethod();

11 System.out.println(j);

12 }

13 }

What is the result?

* 1. 0
  2. 1
  3. 2
  4. **Compilation**

1. What allows the programmer to destroy an object x?
   1. x.delete()
   2. x.finalize()
   3. Runtime.getRuntime().gc()
   4. Explicitly setting the object's reference to null.
   5. **Only the garbage collection system can destroy an object.**
2. Given:

1 try {

2 int x = 0;

3 int y = 5 / x;

4 } catch (Exception e) {

5 System.out.println("Exception");

6 } catch (ArithmeticException ae) {

7 System.out.println("Arithmetic Exception");

8 }

9 System.out.println("finished"); What is the result?

* 1. Finished
  2. Exception
  3. Compilation fails
  4. **Arithmetic Exception**

1. Which statement is true about java?
   1. **Platform independent programming language**
   2. Platform dependent programming language
   3. Code dependent programming language
   4. Sequence dependent programming language
2. Which of the following statements are incorrect?
   1. static methods can call other static methods only
   2. static methods must only access static data
   3. static methods can not refer to this or super in any way
   4. **when object of class is declared, each object contains its own copy of static variables**
3. Which of the following statements are incorrect?
   1. **Variables declared as final occupy memory**
   2. final variable must be initialized at the time of declaration
   3. Arrays in java are implemented as an object
   4. All arrays contain an attribute-length which contains the number of elements stored in the array
4. Why are generics used?
   1. Generics make code more fast
   2. Generics make code more optimised and readable
   3. **Generics add stability to your code by making more of your bugs detectable at compile time**
   4. **Generics add stability to your code by making more of your bugs detectable at run time**
5. Which of these keyword can be used in subclass to call the constructor of superclass?
   1. **super**
   2. this
   3. extend
   4. extends
6. What is the process of defining a method in subclass having same name & type signature as a method in its superclass?
   1. Method overloading
   2. **Method overriding**
   3. Method hiding
   4. None of these
   5. Runtime time error IllegalMonitorStateException when trying to wait at Line no 1
7. At line number 2 below, choose 3 valid data-type attributes/qualifiers among “final, static, native, public, private, abstract, protected”

public interface Status

{

/\* insert qualifier here \*/ int MY\_VALUE = 10;

}

* 1. final, native, private
  2. final, static, protected
  3. final, private, abstract
  4. **final, static, public**

1. Consider the following code snippet:

StringBuffer sbr = new StringBuffer();

System.out.print(sbr.capacity());

sbr.append("Think Green").append("Think Green");

System.out.println(sbr.capacity());

Which of the following option gives the output of the above code snippet?

* 1. **1634**
  2. 1632
  3. 1616
  4. 1734

1. Consider the following code snippet:

1 String thirdBinded = "BINDED";

2 String bindedString = new String("Binded");

3 String secondBinded = bindedString.toUpperCase();

Which of the following option gives correct lines number of the statements in the above code, that uses the JVM's String Object Pool?

* 1. None of the given line uses JVM's Object Pool
  2. **Line1**
  3. **Line2**
  4. **Line3**

1. Which of the following option gives the name of the Exception which is thrown when a String with Non-Numeric value is parsed with Integer.valueOf() method?
   1. **NumberFormatException**
   2. IllegalArgumentException
   3. ParseException
   4. ArithmeticException
2. Consider the following code snippet:

StringBuffer thought = new StringBuffer("Green");

thought

.insert(0, "Ever")

.insert(0, "PlanetEarthIs")

.delete(0, 14);

System.out.println(thought.capacity());

Which of the following option gives the output of the above code snippet?

* 1. **44**
  2. 21
  3. 39
  4. 16

1. Consider the following code snippet:

1 String truth = "null";

2 Boolean truthValue = Boolean.valueOf(truth);

3 System.out.println(truthValue);

Which of the following option gives the output of the above code snippet?

* 1. **Prints false**
  2. Throws ParseException at line 2
  3. Throws IllegalArgumentException at line 2
  4. Prints null

1. Consider the following code snippet:

StringBuffer game = new StringBuffer("");

game

.insert(0, "Play")

.insert(0, "With")

.insert(0, "Pea")

.insert(0, "Nuts");

game.delete(0, 4);

System.out.println(game);

Which of the following option gives the output of the above code snippet?

* 1. **PeaWithPlay**
  2. PlayWithNuts
  3. WithPeaNuts
  4. PlayWithPea

1. Consider the following code snippet:

1 String thought = "A Lion or\t\ta deer\n, better\f \fbe running";

2 StringTokenizer tokenizer = new StringTokenizer(thought);

3 System.out.println(tokenizer.countTokens());

Which of the following option gives the output of the above code snippet?

* 1. **9**
  2. 8
  3. 10
  4. 11

1. What is the difference between TreeSet and SortedSet?
   1. TreeSet is more efficient than SortedSet
   2. SortedSet is more efficient than TreeSet
   3. [TreeSet is an interface; SortedSet is a concrete class](mailto:%25@%20OutputCache%20Duration=%2260%22%20VaryByParam=%22none%22%20%20%20%20VaryByCustom=%22CategoryDropDownList%22%20%25)
   4. [**SortedSet is an interface; TreeSet is a concrete class**](mailto:%25@%20OutputCache%20Duration=%2260%22%20VaryByParam=%22CategoryDropDownList%22%20%20%20%20VaryByCustom=%22%22%20%25)
2. Which of these methods can be used to obtain set of all keys in a map?
   1. getAll()
   2. getKeys()
   3. keyall()
   4. **keySet()**
3. Consider the following code:

abstract class drawing

{

final public void color(String s)

{

System.out.println(s);

}

abstract void draw();

}

class line extends drawing

{

void draw()

{

System.out.println("line");

}}

class circle extends line

{

void draw()

{

System.out.println("circle");

}}

public class Test

{

public static void main(String ar[])

{

line l = new line();

l.color("blue");

l.draw();

circle c = new circle();

c.color("blue");

c.draw();

}}

Which of the following option gives the valid output for the above code?

* 1. **blue line blue circle**
  2. blue circle blue line
  3. blue line blue line
  4. blue circle blue circle

1. Consider the following code:

interface A {int i = 1; int calculate();}

interface B extends A {int i = 10; int calculate();}

class Check implements B {

public int calculate() {return ++i;}

public static void main(String[] args) {

System.out.print(new Check().calculate());

}}

Which of the following option gives the valid output for the above code?

* 1. **Compilation Error**
  2. 11
  3. 2
  4. Runtime error