Java Standard Edition 6 Programmer Certified Professional Exam - Mock Exam V

Section 1: Declarations, Initialization and Scoping

- Develop code that declares classes (including abstract and all forms of nested classes), interfaces, and enums, and includes the appropriate use of package and import statements (including static imports).
- Develop code that declares an interface. Develop code that implements or extends one or more interfaces.
- Develop code that declares an abstract class. Develop code that extends an abstract class.
- Develop code that declares, initializes, and uses primitives, arrays, enums, and objects as static, instance, and local variables. Also, use legal identifiers for variable names.
- Given a code example, determine if a method is correctly overriding or overloading another method, and identify legal return values (including covariant returns), for the method.
- Given a set of classes and superclasses, develop constructors for one or more of the classes. Given a class declaration, determine if a default constructor will be created, and if so, determine the behavior of that constructor. Given a nested or non-nested class listing, write code to instantiate the class.

Section 2: Flow Control

- Develop code that implements an if or switch statement; and identify legal argument types for these statements.
- Develop code that implements all forms of loops and iterators, including the use of for, the enhanced for loop (for-each), do, while, labels, break, and continue; and explain the values taken by loop counter variables during and after loop execution.
- Develop code that makes use of assertions, and distinguish appropriate from inappropriate uses of assertions.
- Develop code that makes use of exceptions and exception handling clauses (try, catch, finally), and declares methods and overriding methods that throw exceptions.
- Recognize the effect of an exception arising at a specified point in a code fragment. Note that the exception may be a runtime exception, a checked exception, or an error.
- Recognize situations that will result in any of the following being thrown: ArrayIndexOutOfBoundsException,ClassCastException, IllegalArgumentException, IllegalStateException, NullPointerException, NumberFormatException, AssertionError, ExceptionInInitializerError, StackOverflowError or NoClassDefFoundError. Understand which of these are thrown by the virtual machine and recognize situations in which others should be thrown programatically.

Section 3: API Contents

- Develop code that uses the primitive wrapper classes (such as Boolean, Character, Double, Integer, etc.), and/or autoboxing & unboxing. Discuss the differences between the String, StringBuilder, and StringBuffer classes.
- Given a scenario involving navigating file systems, reading from files, writing to files, or interacting with the user, develop the correct solution using the following classes (sometimes in combination), from java. io: BufferedReader, BufferedWriter, File, FileReader, FileWriter, PrintWriter, and Console.
- Use standard J2SE APIs in the java.text package to correctly format or parse dates, numbers, and currency values for a specific locale; and, given a scenario, determine the appropriate methods to use if you want to use the default locale or a specific locale. Describe the purpose and use of the java.util. Locale class.
- Write code that uses standard J2SE APIs in the java.util and java.util.regex packages to format or parse strings or streams. For strings, write code that uses the Pattern and Matcher classes and the String.split method. Recognize and use regular expression patterns for matching (limited to: . (dot), * (star), + (plus), ?, \d, \s, \w, [], ()). The use of *, +, and ? will be limited to greedy quantifiers, and the parenthesis operator will only be used as a grouping mechanism, not for capturing content during matching. For streams, write code using the Formatter and Scanner classes and the PrintWriter.format/printf methods. Recognize and use formatting parameters (limited to: %b, %c, %d, %f, %s) in format strings.

Section 4: Concurrency

- Write code to define, instantiate, and start new threads using both java.lang.Thread and java.lang.

Runnable.
- Recognize the states in which a thread can exist, and identify ways in which a thread can transition from one state to another.
- Given a scenario, write code that makes appropriate use of object locking to protect static or instance variables from concurrent access problems.

Section 5: OO Concepts

- Develop code that implements tight encapsulation, loose coupling, and high cohesion in classes, and describe the benefits.
- Given a scenario, develop code that demonstrates the use of polymorphism. Further, determine when casting will be necessary and recognize compiler vs. runtime errors related to object reference casting.
- Explain the effect of modifiers on inheritance with respect to constructors, instance or static variables, and instance or static methods.
- Given a scenario, develop code that declares and/or invokes overridden or overloaded methods and code that declares and/or invokes superclass, or overloaded constructors.
- Develop code that implements "is-a" and/or "has-a" relationships.

Section 6: Collections / Generics

- Given a design scenario, determine which collection classes and/or interfaces should be used to properly implement that design, including the use of the Comparable interface.
- Distinguish between correct and incorrect overrides of corresponding hashCode and equals methods, and explain the difference between == and the equals method.
- Write code that uses the generic versions of the Collections API, in particular, the Set, List, and Map interfaces and implementation classes. Recognize the limitations of the non-generic Collections API and how to refactor code to use the generic versions. Write code that uses the NavigableSet and NavigableMap interfaces.
- Develop code that makes proper use of type parameters in class/interface declarations, instance variables, method arguments, and return types; and write generic methods or methods that make use of wildcard types and understand the similarities and differences between these two approaches.
- Use capabilities in the java.util package to write code to manipulate a list by sorting, performing a binary search, or converting the list to an array. Use capabilities in the java.util package to write code to manipulate an array by sorting, performing a binary search, or converting the array to a list. Use the java.util.Comparator and java.lang.Comparable interfaces to affect the sorting of lists and arrays. Furthermore, recognize the effect of the "natural ordering" of primitive wrapper classes and java.lang. String on sorting.

Section 7: Fundamentals

- Given a code example and a scenario, write code that uses the appropriate access modifiers, package declarations, and import statements to interact with (through access or inheritance) the code in the example.
- Given an example of a class and a command-line, determine the expected runtime behavior.
- Determine the effect upon object references and primitive values when they are passed into methods that perform assignments or other modifying operations on the parameters.
- Given a code example, recognize the point at which an object becomes eligible for garbage collection, determine what is and is not guaranteed by the garbage collection system, and recognize the behaviors of the Object.finalize() method.
- Given the fully-quali fied name of a class that is deployed inside and/or outside a JAR file, construct the appropriate directory structure for that class. Given a code example and a classpath, determine whether the classpath will allow the code to compile successfully.
- Write code that correctly applies the appropriate operators including assignment operators (limited to: =, +=, -=), arithmetic operators (limited to: +, -, *, /, %, ++, --), relational operators (limited to: <, <=, >, >=, ==, !=), the instanceof operator, logical operators (limited to: &, |, ^, !, &&, ||), and the conditional operator (? :), to produce a desired result. Write code that determines the equality of two objects or two primitives

Exam A

QUESTION 1
Given:

```
11. public static void test(String str) {
12.    if(str == null | str.length() == 0) {
13.       System.out.println("String is empty");
14.    } else {
15.       System.out.println("String is not empty");
16.    }
17. }
```

And the invocation:

```
31. test(null);
```

What is the result?

A. An exception is thrown at runtime.
B. "String is empty" is printed to output.
C. Compilation fails because of au error in line 12.
D. "String is not empty" is printed to output.

Correct Answer: A

QUESTION 2
Given:

```
11. public static void main(String[] args) {
12.    try {
13.       args=null;
14.       args[0] = "test";
15.       System.out.println(args[0]);
16.    } catch (Exception ex) {
17.         System.out.println("Exception");
18.    } catch (NullPointerException npe) {
19.         System.out.println("NullPointerException");
20.    }
21. }
```

What is the result?

A. test
B. Exception
C. Compilation fails.
D. NullPointerException

Correct Answer: C

QUESTION 3
Given:

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

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

A. final
B. static
C. native

D. public
E. private
F. abstract
G. protected

Correct Answer: ABD

QUESTION 4
Which four are true? (Choose four.)

A. Has-a relationships should never be encapsulated.
B. Has-a relationships should be implemented using inheritance.
C. Has-a relationships can be implemented using instance variables.
D. Is-a relationships can be implemented using the extends keyword.
E. Is-a relationships can be implemented using the implements keyword.
F. The relationship between Movie and Actress is an example of an is-a relationship.
G. An array or a collection can be used to implement a one-to-many has-a relationship.

Correct Answer: CDEG

QUESTION 5
Which statements are true about comparing two instances of the same class, given that the equals() and hashCode() methods have been properly overridden? (Choose two)

A. If the equals() method returns true, the haschCode() comparison == might return false.
B. If the equals() method returns false, the haschCode() comparison == might return true.
C. If the hashCode() comparison == returns true, the equals() method must return true.
D. If the hashCode() comparison == returns true, the equals() method might return true.
E. If the hashCode() comparison != returns true, the equals() method might return true.

Correct Answer: BD

QUESTION 6
Given:

```
1. public class Person {
2.    private String name;
3.    public Person(String name) { this.name = name; }
4.    public boolean equals(Person p) {
5.       return p.name.equals(this.name);
6.    }
7. }
```

Which is true?

A. The equals method does NOT properly override the Object.equals method.
B. Compilation fails because the private attribute p.name cannot be accessed in line 5.
C. To work correctly with hash-based data structures, this class must also implement the hashCode method.
D. When adding Person objects to a java.util.Set collection, the equals method in line 4 will prevent duplicates.

Correct Answer: A

QUESTION 7
Given:

```
11. public class Person {
12.    private name;
```

```
13.    public Person(String name) {
14.       this.name = name;
15.    }
16.    public int hashCode() {
17.       return 420;
18.    }
19. }
```

Which is true?

A. The time to find the value from HashMap with a Person key depends on the size of the map.
B. Deleting a Person key from a HashMap will delete all map entries for all keys of type Person.
C. Inserting a second Person object into a HashSet will cause the first Person object to be removed as a duplicate.
D. The time to determine whether a Person object is contained in a HashSet is constant and does NOT depend on the size of the map.

Correct Answer: A

QUESTION 8
Given:

```
11. public class Person {
12.    private name;
13.    public Person(String name) {
14.       this.name = name;
15.    }
16.    public boolean equals(Object o) {
17.       if( !o instanceof Person ) return false;
18.       Person p = (Person) o;
19.       return p.name.equals(this.name);
20.    }
21. }
```

Which is true?

A. Compilation fails because the hashCode method is not overridden.
B. A HashSet could contain multiple Person objects with the same name.
C. All Person objects will have the same hash code because the hashCode method is not overridden.
D. If a HashSet contains more than one Person object with name = "Fred", then removing another Person, also with name ="Fred", will remove them all.

Correct Answer: B

QUESTION 9
Given:

```
11. public class Person {
12.    private String name, comment;
13.    private int age;
14.    public Person(String n, int a, String c) {
15.       name = n; age = a; comment = c;
16.    }
17.    public boolean equals(Object o) {
18.       if(!(o instanceof Person)) return false;
19,       Person p = (Person)o;
20.       return age == p.age && name.equals(p.name);
21.    }
22. }
```

What is the appropriate definition of the hashCode method in class Person?

A. return super.hashCode();

B. return name.hashCode() + age * 7;
C. return name.hashCode() + comment.hashCode() / 2;
D. return name.hashCode() + comment.hashCode() / 2 - age * 3;

Correct Answer: B

QUESTION 10
Given:

```
11. public class Key {
12.    private long id1;
13.    private long 1d2;
14.
15.    // class Key methods
16. }
```

A programmer is developing a class Key, that will be used as a key in a standard java.util.HashMap. Which two methods should be overridden to assure that Key works correctly as a key? (Choose two.)

A. public int hashCode()
B. public boolean equals(Key k)
C. public int compareTo(Object o)
D. public boolean equals(Object o)
E. public boolean compareTo(Key k)

Correct Answer: AD

QUESTION 11
Given:

```
1. public interface A {
2.    String DEFAULT_GREETING = "Hello World";
3.    public void method1();
4. }
```

A programmer wants to create an interface called B that has A as its parent. Which interface declaration is correct?

A. public interface B extends A {}
B. public interface B implements A {}
C. public interface B instanceOf A {}
D. public interface B inheritsFrom A {}

Correct Answer: A

QUESTION 12
Given:

```
package test;

class Target {
        public String name = "hello";
}
```

What can directly access and change the value of the variable name?

A. any class
B. only the Target class
C. any class in the test package
D. any class that extends Target

Correct Answer: C

QUESTION 13
Given two files:

```
1.    package pkgA;
2.    public class Foo {
3.            int a = 5;
4.            protected int b = 6;
5.    }
```

```
1.    package pkgB;
2.    import pkgA.*;
3.    public class Fiz extends Foo {
4.             public static void main(String[] args) {
5.                    Foo f = new Foo();
6.                    System.out.print(" " + f.a);
7.                    System.out.print(" " + f.b);
8.                    System.out.print(" " + new Fiz().a);
9.                    System.out.println(" " + new Fiz().b);
10.           }
11.  }
```

What is the result? (Choose all that apply)

A. 5 6 5 6
B. 5 6 followed by an exception
C. Compilation fails with an error on line 6
D. Compilation fails with an error on line 7
E. Compilation fails with an error on line 8
F. Compilation fails with an error on line 9

Correct Answer: CDE

QUESTION 14
Given two files:

```
1. package pkgA;
2. public class Foo {
3.     int a = 5;
4.     protected int b = 6;
5.     public int c = 7;
6. }
```

```
3. package pkgB;
4. import pkA.*;
5. public class Baz {
6.    public static void main(String[] args) {
7.          Foo f = new Foo();
8.          System.out.print(" " + f.a);
9.          System.out.print(" " + f.b);
10.         System.out.print(" " + f.c);
11.   }
12.}
```

What is the result? (Choose all that apply)

A. 5 6 7
B. 5 followeb by an exception
C. Compilation fails with an error on line 7
D. Compilation fails with an error on line 8
E. Compilation fails with an error on line 9

F. Compilation fails with an error on line 10

Correct Answer: DE

QUESTION 15
A programmer has an algorithm that requires a java.util.List that provides an efficient implementation of add (0,object), but does NOT need to support quick random access. What supports these requirements?

A. java.util.Queue
B. java.util.ArrayList
C. java.util.LinearList
D. java.util.LinkedList

Correct Answer: D

QUESTION 16
Given:

```
package com.sun.scjp;
public class Geodetics {
        public static final double DIAMETER = 12756.32; // kilometers
}
```

Which two correctly access the DIAMETER member of the Geodetics class? (Choose two.)

A.
```
import com.sun.scjp.Geodetics;
public class TerraCarta {
     public double halfway() {
        return Geodetics.DIAMETER/2.0;
     }
}
```
B.
```
import static com.sun.scjp.Geodetics;
public class TerraCarta {
    public double halfway() {
       return DIAMETER/2.0;
    }
}
```
C.
```
import static com.sun.scjp.Geodetics. *;
public class TerraCarta {
      public double halfway() {
          return DIAMETER/2.0;
      }
}
```
D.
```
package com.sun.scjp;
public class TerraCarta {
      public double halfway() {
          return DIAMETER/2.0;
      }
}
```

Correct Answer: AC

QUESTION 17
Given:

```
public abstract interface Frobnicate
{
     public void twiddle(String s);
}
```

Which is a correct class? (Choose all that apply)

A.
```
public abstract class Frob implements Frobnicate {
     public abstract void twiddle(String s) { }
}
```
B.
```
public abstract class Frob implements Frobnicate
{ }
```
C.
```
public class Frob extends Frobnicate {
    public void twiddle(Integer i) { }
}
```
D.
```
public class Frob implements Frobnicate {
     public void twiddle(Integer i) { }
}
```
E.
```
public class Frob implements Frobnicate {
     public void twiddle(Integer s) { }
}
```

Correct Answer: B

QUESTION 18
Which collection class(es) allows you to grow or shrink its size and provides indexed access to its elements, but whose methods are not synchronized?

A. java.util.HashSet
B. java.util.LinkedHashSet
C. java.util.List
D. java.util.ArrayList
E. java.util.Vector
F. java.util.PriorityQueue

Correct Answer: D

QUESTION 19
Given:

```
1. public  class Electronic implements Device
    {  public void doIt() { } }
2.
3. abstract class Phone1 extends Electronic {}
4.
5. abstract class Phone2 extends Electronic
    { public void doIt(int x){ } }
6.
7. class Phone3 extends Electronic implements Device
    { public void doStuff() { } }
8.
9. interface Device { public void doIt(); }
```

What is the result? (Choose all that apply)

A. Compilation suceeds
B. Compilation fails with an error on line 1
C. Compilation fails with an error on line 3
D. Compilation fails with an error on line 5
E. Compilation fails with an error on line 7
F. Compilation fails with an error on line 9

Correct Answer: A

QUESTION 20
Given:

```
import java.util.*;
```

```
class MapEQ {

       public static void main(String[] args) {
           Map<ToDos, String> m = new HashMap<ToDos, String>();
           ToDos t1 = new ToDos("Monday");
           ToDos t2 = new ToDos("Monday");
           ToDos t3 = new ToDos("Tuesday");
           m.put(t1, "doLaundry");
           m.put(t2, "payBills");
           m.put(t3, "cleanAttic");
           System.out.println(m.size());
      }
}

class ToDos {
           String day;
           ToDos(String d) { day = d; }
           public boolean equals(Object o) {
                return ((ToDos)o).day == this.day;
           }
           //public int hashCode() { return 9; }
}
```

Which is correct? (Choose two)

A. As the code stands it will not compile
B. As the code stands the output will be 2
C. As the code stands the output will be 3
D. If the hashCode() method is uncommented the output will be 2
E. If the hashCode() method is uncommented the output will be 3
F. If the hashCode() methods is uncommented the code will not compile

Correct Answer: CD

QUESTION 21
Given:

```
1.   class Pizza {
2.      java.util.ArrayList toppings;
3.      public final void addTopping(String topping) {
4.         toppings.add(topping);
5.      }
6.   }
7.   public class PepperoniPizza extends Pizza {
8.      public void addTopping(String topping) {
9.         System.out.println("Cannot add Toppings");
10.    }
11.    public static void main(String[] args) {
12.       Pizza pizza = new PepperoniPizza();
13.       pizza.addTopping("Mushrooms");
14.    }
15. }
```

What is the result?

A. Compilation fails.
B. Cannot add Toppings
C. The code runs with no output.
D. A NullPointerException is thrown in Line 4.

Correct Answer: A

QUESTION 22

Given:

```
1. import java.util.*;
2. public class Example {
3.    public static void main(String[] args) {
4.       // insert code here
5.       set.add(new Integer(2));
6.       set.add(new Integer(1));
7.       System.out.println(set);
8.    }
9. }
```

Which code, inserted at line 4, guarantees that this program will output [1, 2]?

A. Set<Integer> set = new TreeSet<Integer>();
B. Set<Integer> set = new HashSet<Integer>();
C. Set<Integer> set = new SortedSet<Integer>();
D. List<Integer> set = new SortedList<Integer>();
E. Set<Integer> set = new LinkedHashSet<Integer>();

Correct Answer: A

QUESTION 23
Given:

```
import java.util.*;

class Dog {
        int size;
        Dog (int s) {
           size = s;
        }
}
public class FirstGrade {
        public static void main(String[] args) {
                TreeSet<Integer> i = new TreeSet<Integer>();
                TreeSet<Dog> d = new TreeSet<Dog>();

                d.add(new Dog(1));
                d.add(new Dog(2));
                d.add(new Dog(1));
                i.add(1);
                i.add(2);
                i.add(1);
                System.out.println(d.size() + " " + i.size());
}
```

What is the result?

A. 1 2
B. 2 2
C. 2 3
D. 3 2
E. 3 3
F. Compilation fails
G. An exception is thrown at runtime

Correct Answer: G

QUESTION 24
Given:

```
1.   import java.util.*;
```

```
2.   public class WrappedString {
3.      private String s;
4.      public WrappedString(String s) { this.s = s; }
5.      public static void main(String[] args) {
6.         HashSet<Object> hs = new HashSet<Object>();
7.         WrappedString ws1 = new WrappedString("aardvark");
8.         WrappedString ws2 = new WrappedString("aardvark");
9.         String s1 = new String("aardvark");
10.       String s2 = new String("aardvark");
11.       hs.add(ws1); hs.add(ws2); hs.add(s1); hs.add(s2);
12.       System.out.println(hs.size());
13.  } }
```

What is the result?

A. 0
B. 1
C. 2
D. 3
E. 4
F. Compilation fails.
G. An exception is thrown at runtime.

Correct Answer: D

QUESTION 25
Given:

```
10. public class Hello {
11.    String title;
12.    int value;
13.    public Hello() {
14.       title += " World";
15.    }
16.    public Hello(int value) {
17.       this.value = value;
18.       title = "Hello ";
19.       Hello();
20.    }
21. }
```

and:

```
30. Hello c = new Hello(5);
31. System.out.println(c.title);
```

What is the result?

A. Hello
B. Hello World
C. Compilation fails.
D. Hello World 5
E. The code runs with no output.
F. An exception is thrown at runtime.

Correct Answer: C

QUESTION 26
Given:

```
10. public class SuperCalc {
11.    protected static int multiply(int a, int b) { return a * b; }
```

```
12. }
```

and:

```
20. public class SubCalc extends SuperCalc {
21.    public static int multiply(int a, int b) {
22.       int c = super.multiply(a, b);
23.       return c;
24.    }
25. }
```

and:

```
30. SubCalc sc = new SubCalc();
31. System.out.println(sc.multiply(3,4));
32. System.out.println(SubCalc.multiply(2,2));
```

What is the result?

A. 12
   4
B. The code runs with no output.
C. An exception is thrown at runtime.
D. Compilation fails because of an error in line 21.
E. Compilation fails because of an error in line 22.
F. Compilation fails because of an error in line 31.

Correct Answer: E

QUESTION 27
Click the Exhibit button.

Which code, inserted at line 7, completes the Salesperson constructor?

Exhibit:

```
1. public class Employee {
2.    String name;
3.    double baseSalary;
4.    Employee(String name, double baseSalary) {
5.       this.name = name;
6.       this.baseSalary = baseSalary;
7.    }
8. }
```

And:

```
1. public class Salesperson extends Employee {
2.    double commission;
3.    public Salesperson(String name, double baseSalary,
4.       double commission) {
5.          // insert code here
6.    }
7. }
```

A. this.commission = commission;

B. super();
commission = commission;
C. this.commission = commission;
super();
D. super(name, baseSalary);
this.commission = commission;
E. super();
this.commission = commission;
F. this.commission = commission;
super(name, baseSalary);

Correct Answer: D

QUESTION 28
Click the Exhibit button.

What two must the programmer do to correct the compilation errors? (Choose two.)

Exhibit:

```
1.  public class Car {
2.    private int wheelCount;
3.    private String vin;
4.    public Car(String vin) {
5.      this.vin = vin;
6.      this.wheelCount = 4;
7.    }
8.    public String drive() {
9.      return "zoom-zoom";
10.   }
11.   public String getInfo() {
12.     return "VIN: "+ vin + "wheels: "+ wheelCount;
13.   }
14. }
```

And:

```
1. public class MeGo extends Car {
2.   public MeGo(String vin) {
3.     this.wheelCount = 3;
4.   }
5. }
```

A. insert a call to this() in the Car constructor

B. insert a call to this() in the MeGo constructor
C. insert a call to super() in the MeGo constructor
D. insert a call to super(vin) in the MeGo constructor
E. change the wheelCount variable in Car to protected
F. change line 3 in the MeGo class to super.wheelCount = 3;

Correct Answer: DE

QUESTION 29
Given a file GrizzlyBear.java:

```
1. package animals.mammals;
2.
3. public class GrizzlyBear extends Bear {
4.    void hunt() {
5.       Salmon s = findSalmon();
6.       s.consume();
7.    }
8. }
```

and another file, Salmon.java:

```
1. package animals.fish;
2.
3. public class Salmon extends Fish {
4.    void consume() { /* do stuff */ }
5. }
```

Assume both classes are defined in the correct directories for theft packages, and that the Mammal class correctly defines the findSalmon() method.
Which two changes allow this code to compile correctly? (Choose two.)

A. add public to the start of line 4 in Salmon.java
B. add public to the start of line 4 in GrizzlyBear.java
C. add import animals.mammals.*; at line 2 in Salmon.java
D. add import animals.fish.*; at line 2 in GrizzlyBear.java
E. add import animals.fish.Salmon.*; at line 2 in GrizzlyBear.java
F. add import animals.mammals.GrizzlyBear.*; at line 2 in Salmon.java

Correct Answer: AD

QUESTION 30
Given:

```
1. public class A {
2.      public void doit() {
3.      }
4.      public String doit() {
5.         return "a";
6.      }
7.      public double doit(int x) {
8.         return 1.0;
9.      }
10. }
```

What is the result?

A. An exception is thrown at runtime.
B. Compilation fails because of an error in line 7.
C. Compilation fails because of an error in line 4.
D. Compilation succeeds and no runtime errors with class A occur.

Correct Answer: C

QUESTION 31
Given:

```
1. public class A {
2.      public String doit(int x, int y) {
3.          return "a";
4.      }
5.
6.      public String doit(int... vals) {
7.             return "b";
8.       }
9. }
```

Given:

```
25. A a=new A();
26. System.out.println(a.doit(4, 5));
```

What is the result?

A. Line 26 prints "a" to System.out.
B. Line 26 prints 'b" to System.out.
C. An exception is thrown at line 26 at runtime.
D. Compilation of class A will fail due to an error in line 6.

Correct Answer: A

QUESTION 32
Given:

```
   class TestA {
           public void start() {
                 System.out.println("TestA");
           }
   }
   public class TestB extends TestA {
           public void start() {
                 System.out.println("TestB");
            }
           public static void main(String[] args) {
                ((TestA)new TestB()).start();
           }
  }
```

What is the result?

A. TestA
B. TestB
C. Compilation fails.
D. An exception is thrown at runtime.

Correct Answer: B

QUESTION 33
Given:

```
class One {
     public One() { System.out.print(1); }
 }

 class Two extends One {
     public Two() { System.out.print(2); }
```

```
}

class Three extends Two {
      public Three() { System.out.print(3); }
}

public class Numbers{
     public static void main(String[] args) {
        new Three();
     }
}
```

What is the result when this code is executed?

A. 1
B. 3
C. 123
D. 321
E. The code runs with no output.

Correct Answer: C

QUESTION 34
Given:

```
public class Plant {
    private String name;
    public Plant(String name) { this.name = name; }
    public String getName() { return name; }
}

public class Tree extends Plant {
    public void growFruit() { }
    public void dropLeaves() { }
}
```

Which is true?

A. The code will compile without changes.
B. The code will compile if public Tree() { Plant(); } is added to the Tree class.
C. The code will compile if public Plant() { Tree(); } is added to the Plant class.
D. The code will compile if public Plant() { this(”fern”); } is added to the Plant class.
E. The code will compile if public Plant() { Plant(”fern”); } is added to the Plant class.

Correct Answer: D

QUESTION 35
Given:

```
class One {
       public One foo() { return this; }
}

class Two extends One {
      public One foo() { return this; }
}

class Three extends Two {
     // insert method here
}
```

Which two methods, inserted individually, correctly complete the Three class? (Choose two.)

A. public void foo() { }
B. public int foo() { return 3; }
C. public Two foo() { return this; }
D. public One foo() { return this; }
E. public Object foo() { return this; }

Correct Answer: CD

QUESTION 36
Given:

```
class Person {
     String name = "No name';
     public Person(String nm) { name = nm; }
}

class Employee extends Person {
     String emplD = "0000";
     public Employee(String id) { empID = id; }    <----
}

public class EmployeeTest {
     public static void main(String[] args) {
           Employee e = new Employee("4321");
           System.out.println(e.empID);
     }
}
```

What is the result?

A. 4321
B. 0000
C. An exception is thrown at runtime.
D. Compilation fails because of an error the marked instruction (<----)

Correct Answer: D

QUESTION 37
Given:

```
1. public class A {
2.
3.     private int counter = 0;
4.
5.     public static int getInstanceCount() {
6.              return counter;
7.     }
8.
9.     public A() {
10.           counter++;
11.    }
12.
13. }
```

Given this code from Class B:

```
25. A a1 =new A();
26. A a2 =new A();
27. A a3 =new A();
28. System.out.printIn(A.getInstanceCount() );
```

What is the result?

A. Compilation of class A fails.
B. Line 28 prints the value 3 to System.out.
C. Line 28 prints the value 1 to System.out.
D. A runtime error occurs when line 25 executes.
E. Compilation fails because of an error on line 28.

Correct Answer: A

QUESTION 38
Given:

```
class ClassA {}
class ClassB extends ClassA {}
class ClassC extends ClassA {}
```

and:

```
ClassA p0 = new ClassA();
ClassB p1 = new ClassB();
ClassC p2 = new ClassC();
ClassA p3 = new ClassB();
ClassA p4 = new ClassC();
```

Which three are valid? (Choose three.)

A. p0 = p1;
B. p1 =p2;
C. p2 = p4;
D. p2 = (ClassC)p1;
E. p1 = (ClassB)p3;
F. p2 = (ClassC)p4;

Correct Answer: AEF

QUESTION 39
Given:

```
class Foo {
      static void alpha() { /* more code here */ }
      void beta() { /* more code here */ }
}
```

Which two are true? (Choose two.)

A. Foo.beta() is a valid invocation of beta().
B. Foo.alpha() is a valid invocation of alpha().
C. Method beta() can directly call method alpha().
D. Method alpha() can directly call method beta().

Correct Answer: BC

QUESTION 40
Given:

```
public class Base {
   public static final String FOO = “foo”;

  public static void main(String[] args) {
       Base b = new Base();
       Sub s = new Sub();
       System.out.print(Base.FOO);
```

```
        System.out.print(Sub.FOO);
        System.out.print(b.FOO);
        System.out.print(s.FOO);
        System.out.print(((Base)s).FOO);
    }
}

 class Sub extends Base {
      public static final String FOO="bar";
}
```

What is the result?

A. foofoofoofoofoo
B. foobarfoobarbar
C. foobarfoofoofoo
D. foobarfoobarfoo
E. barbarbarbarbar
F. foofoofoobarbar
G. foofoofoobarfoo

Correct Answer: D

QUESTION 41
Given:

```
class Alpha
{
    public void foo(String… arg) {System.out.println(”Alpha.foo”); }
    public void bar(String arg) { System.out.println(”Alpha.bar”);}
}

class Beta extends Alpha
{
    public void foo(String… arg) {System.out.println(”Beta.foo”); }
    public void bar(String arg) { System.out.println(”Beta.bar”);}
}

public class Test {

    public static void main(String a[])
    {
        Alpha alpha = new Beta();
        Beta beta = (Beta) alpha;

        alpha.foo("sabbir");
        alpha.bar("Jhon");
        beta.foo("sakera");
        beta.bar("Anjellina");

    }
}
```

What is the ouput?

A. Alpha.foo
   Alpha.bar
   Beta.foo
   Beta.bar
B. Beta.foo
   Beta.bar
   Alpha.foo
   Alpha.bar

C. Beta.foo
Beta.bar
Beta.foo
Beta.bar

D. Alpha.foo
Alpha.bar
Alpha.foo
Alpha.bar

Correct Answer: C

QUESTION 42
Given:

```
public interface A {
    public void doSomething(String thing);
}

public class AImpl implements A {
   public void doSomething(String msg)// line 2
   { }
}

public class B
{
    public A doit() {
         return new AImpl();
    }
    public String execute(){
        return "";
    }
}

public class C extends B
{
       public AImpl doit() //line 2
       {
              return new AImpl();
       }

      public Object execute() //line 6
      {
              return "";
      }
}
```

Which will be true …

A. Compilation will succeed for all classes and interfaces
B. Compilation of class C will fail because of an error in line 2
C. Compilation of class C will fail because of an error in line 6
D. Compilation of class AImpl will fail because of an error in line 2

Correct Answer: C

QUESTION 43
Given these classes in different files:

```
package xcom;
public class Useful {
    int increment(int x) {
    return ++x;
    }
}
```

```
import xcom.*; // line 1
class Needy3 {
   public static void main(String[] args) {
        xcom.Useful u = new xcom.Useful();  // line 2
        System.out.println(u.increment(5));
   }
}
```

Which statements are true? (Choose all that apply)

A. The output is 0.
B. The output is 5.
C. The output is 6.
D. Compilation fails.
E. The code compiles if line 1 is removed.
F. The code compiles if line 2 is changed to read
   Useful u = new Useful();

Correct Answer: D

QUESTION 44
Given two files:

```
package xcom;
public class Stuff {
    public static final int MY_CONSTANT = 5;
    public static int doStuff(int x) {
         return (x++)*x;
    }
}

import xcom.Stuff.*;
import java.lang.System.out;
class User {
   public static void main(String[] args) {
      new User().go();
   }
   void go() {
      out.println(doStuff(MY_CONSTANT));
   }
}
```

What is the result?

A. 25
B. 30
C. 36
D. Compilation fails
E. An exception is thrown at runtime

Correct Answer: D

QUESTION 45
Given this three files:

```
package xcom;
public class A {
   // insert code here
}


package xcom;
public class B extends A {
```

```
   public void doB() {
      System.out.println("B.doB");
   }
}

import xcom.B;
class TestXcom {
    public static void main(String[] args) {
       B b = new B();
       b.doB();
       b.go();
    }
}
```

Which code, inserted at // insert code here will allow all three files to compile? (Choose all that apply.)

A.
```
void go() {
        System.out.println("a.go");
   }
```
B.
```
public void go() {
        System.out.println("a.go");
   }
```
C.
```
private void go() {
        System.out.println("a.go");
   }
```
D.
```
protected void go() {
        System.out.println("a.go");
   }
```
E. None of these options will allow the code to compile.

Correct Answer: B

QUESTION 46
Given:

```
class Top {
     public Top(String s) {
            System.out.println("B");
     }
}

public class Bottom2 extends Top {
     public Bottom2(String s) {
             System.out.println("D");
     }
     public static void main(String[] args) {
             new Bottom2("C");
             System.out.println(" ");
     }
}
```

What is the result?

A. BD
B. DB
C. BDC
D. DBC
E. Compilation fails

Correct Answer: E

QUESTION 47
Given:

```
class Clidder {
       private final void flipper()  {
            System.out.println("Clidder");
       }
}

public class Clidlet extends Clidder {
        public final void flipper() {
              System.out.println("Clidlet");
        }
        public static void main(String[] args) {
               new Clidet().flipper();
        }
}
```

What is the result?

A. Clidlet
B. Clidder
C. Clidder
   Clidlet
D. Clidlet
   Clidder
E. Compilation fails

Correct Answer: A

QUESTION 48
Given:

```
3.    class Dog {
4.             public void bark() { System.out.print("woof "); }
5.    }
6.    class Hound extends Dog {
7.             public void sniff() { System.out.print("sniff"); }
8.             public void bark() { System.out.pint("how1 "); }
9.    }
10.  public class DogShow {
11.          public static void main(String[] args) { new DogShow().go(); }
12.          void go() {
13.                 new Hound().bark();
14.                 ((Dog) new Hound()).bark();
15.                 ((Dog) new Hound()).sniff();
16.          }
17.  }
```

What is the result? (Chosse all that apply)

A. how1 how1 sniff
B. how1 woof sniff
C. how1 how1 followed by an exception
D. how1 woof followed by an exception
E. Compilation fails with an error at line 14
F. Compilation fails with an error at line 15

Correct Answer: F

QUESTION 49
Given:

```
3.  public class Tenor extends Singer {
4.             public static String sing() { return "fa"; }
```

```
5.            public static void main(String[] args) {
6.                    Tenor t = new Tenor();
7.                    Singer s = new Tenor();
8.                    System.out.println(t.sing() + " " + s.sing());
9.             }
10. }
11. class Singer { public static String sing() { return "la"; } }
```

What is the result?

A. fa fa
B. fa la
C. la la
D. Compilation fails
E. An exception is thrown at runtime

Correct Answer: B

QUESTION 50
Given:

```
3.  class Alpha {
4.         static String s = " ";
5.         protected Alpha() { s += "alpha "; }
6.  }
7.  class SubAlpha extends Alpha {
8.          private SubAlpha() { s += "sub "; }
9.  }
10.public class SubSubAlpha extends Alpha {
11.         private SubSubAlpha() { s += "subsub "; }
12.         public static void main(String[] args) {
13.                 new SubSubAlpha();
14.                 System.out.println(s);
15.         }
16. }
```

What is the result?

A. subsub
B. sub subsub
C. alpha subsub
D. alpha sub subsub
E. Compilation fails
F. An exception is thrown at runtime

Correct Answer: C

QUESTION 51
Given:

```
3.  class Mammal {
4.         String name = "furry ";
5.         String makeNoise() { return "generic noise"; }
6.  }
7.  class Zebra extends Mammal {
8.          String name = "stripes ";
9.          String makeNoise() { return "bray"; }
10. }
11. public class ZooKeeper {
12.         public static void main(String[] args) { new ZooKeeper().go(); }
13.         void go() {
14.                 Mammal m = new Zebra();
15.                 System.out.println(m.name + m.makeNoise());
```

```
16.          }
17. }
```

What is the result?

A. furry bray
B. stripes bray
C. furry generic noise
D. stripes generic noise
E. Compilation fails
F. An exception is thrown at runtime

Correct Answer: A

QUESTION 52
Given:

```
interface DeclareStuff
{
    public static final int EASY = 3;
    void doStuff(int t);
}

class TestDeclare implements DeclareStuff
{
    public static void main(String[] args) {
        int x = 5;
        new TestDeclare().doStuff(++x);
    }

    public void doStuff(int s) {
        s += EASY + ++s;
        System.out.println(s);
    }
}
```

What is the result?

A. s 14
B. s 16
C. s 10
D. Compilation fails
E. An exception is thrown at runtime.

Correct Answer: B

QUESTION 53
Given:

```
1.  class SuperClass {
2.    public A getA() {
3.      return new A();
4.    }
5.  }
6.  class SubClass extends SuperClass {
7.    public B getA(){
8.      return new B();
9.    }
10. }
```

Which statement is true?

A. Compilation will succeed if A extends B.

B. Compilation will succeed if B extends A.
C. Compilation will always fail because of an error in line7.
D. Compilation will always fail because of an error in line8.

Correct Answer: B

QUESTION 54
A class games.cards.Poker is correctly defined in the jar file Poker.jar.
A user wants to execute the main method of Poker on a UNIX system using the command:

```
java games.cards.Poker
```

What allows the user to do this?

A. put Poker.jar in directory /stuff/java, and set the CLASSPATH to include /stuff/java
B. put Poker.jar in directory /stuff/java, and set the CLASSPATH to include /stuff/java/*.jar
C. put Poker.jar in directory /stuff/java, and set the CLASSPATH to include /stuff/java/Poker.jar
D. put Poker.jar in directory /stuff/java/games/cards, and set the CLASSPATH to include /stuff/java
E. put Poker.jar in directory /stuff/java/games/cards, and set the CLASSPATH to include /stuffijava/*.jar
F. put Poker.jar in directory /stuff/java/games/cards, and set the CLASSPATH to include /stuff/java/Poker.jar

Correct Answer: C

QUESTION 55
A developer is creating a class Book that needs to access class Paper.
The Paper class is deployed in a JAR named myLib.jar.

Which three, taken independently, will allow the developer to use the Paper class while compiling the Book class? (Choose three.)

A. The JAR file is located at $JAVA_HOME/jre/classes/myLib.jar.
B. The JAR file is located at $JAVA_HOME/jre/lib/ext/myLib.jar.
C. The JAR file is located at /foo/myLib.jar and a classpath environment variable is set that includes /foo/myLib.jar/Paper.class
D. The JAR file is located at /foo/myLib.jar and a classpath environment variable is set that includes /foo/myLib.jar.
E. The JAR file is located at /foo/myLib.jar and the Book class is compiled using javac -cp /foo/myLib.jar/Paper Book.java.
F. The JAR file is located at /foo/myLib.jar and the Book class is compiled using javac -d /foo/myLib.jar Book.java.
G. The JAR file is located at /foo/myLib.jar and the Book class is compiled using javac -classpath /foo/myLib.jar Book.java.

Correct Answer: BDG

QUESTION 56
Given:

```
1. package com.company.application;
2.
3. public class MainClass {
4.    public static void main(String[] args) { }
5. }
```

And MainClass exists in the /apps/com/company/application directory.

Assume the CLASSPATH environment variable is set to "." (current directory). Which two java commands entered at the command line will run MainClass?
(Choose two.)

A. java MainClass if run from the /apps directory
B. java com.company.application.MainClass if run from the /apps directory
C. java -classpath /apps com.company.application.MainClass if run from any directory
D. java -classpath . MainClass if run from the /apps/com/company/application directory
E. java -classpath /apps/com/company/application:. MainClass if run from the /apps directory
F. java com.company.application.MainClass if run from the /apps/com/company/application directory

Correct Answer: BC

QUESTION 57
Given:

```
11. String test = 'a1b2c3";
12. String[] tokens = test.split("\\d");
13. for(String s: tokens) System.out.print(s + " ");
```

What is the result?

A. a b c
B. 1 2 3
C. a1b2c3
D. a1 b2 c3
E. Compilation fails
F. The code runs with no output.
G. An exception is thrown at runtime.

Correct Answer: A

QUESTION 58
Given the following code:

```
public class TokenizingUsingString {

     public static void main(String[] args) {
         String[] sa = "a1a2a34a56789a".split("a");
         System.out.println("Number of tokens generated: " + sa.length);
         for (String s : sa) {
             System.out.println(s);
         }
     }
 }
```

What is the output?

A. Compilation fails
B. Runtime exception
C. Number of tokens generated: 6

   1
   2
   34
   56789

D. Number of tokens generated: 5

   1
   2
   34

56789

E. Number of tokens generated: 4
1
2
34
56789

Correct Answer: D

QUESTION 59
Given:

```
import java.util.regex.*;

class Regex2 {
      public static void main(String[] args) {
                Pattern p = Pattern.compile(args[0]);
                Matcher m = p.matcher(args[1]);
                boolean b = false;
                while(b = m.find()) {
                      System.out.print(m.start() + m.group());
                }
       }
}
```

And the command line:
`java Regex2 "\d*" ab34ef`

What is the result?

A. 234
B. 334
C. 2334
D. 0123456
E. 01234456
F. 12334567
G. Compilation

Correct Answer: E

QUESTION 60
Given:

```
11. String test = "This is a test";
12. String[] tokens = test.split("\s");
13. System.out.println(tokens.length);
```

What is the result?

A. 0
B. 1
C. 4
D. Compilation fails.
E. An exception is thrown at runtime.

Correct Answer: D