Question 1.
Given:

```
import java.util.*;
public class AndOver {
public static void main(String[] args) {
       List g = new ArrayList();
       g.add(new Gaited("Eyra"));
       g.add(new Gaited("Vafi"));
       g.add(new Gaited("Andi"));
       Iterator i2 = g.iterator();
       while(i2.hasNext()) {
              System.out.print(i2.next().name + " ");
       }
} }

class Gaited {
       public String name;
       Gaited(String n) { name = n; }
}
```

What is the result?
Select 1 option:

A. Vafi Andi
B. Andi Eyra Vafi
C. Andi Vafi Eyra
D. Eyra Vafi Andi
E. Compilation fails.
F. The output order is unpredictable.

Question 2.
Given:

```
class Grab {
       static int x = 5;
       synchronized void adjust(Grab y) {
              System.out.print(x-- + " ");
              y.view(y);
       }

       synchronized void view(Grab z) { if(x > 0) z.adjust(z); }
}

public class Grapple implements Runnable {
       static Thread t1;
       static Grab g, g2;
       public void run() {
              if(Thread.currentThread().getId() == t1.getId()) g.adjust(g2);
              else g2.view(g);
       }

       public static void main(String[] args) {
              g = new Grab();
              g2 = new Grab();
              t1 = new Thread(new Grapple());
              t1.start();
              new Thread(new Grapple()).start();
       }
}
```

Which are true?
Select 3 options:

A. Compilation fails
B. The output could be 5 4 3 2 1
C. The output could be 5 4 3 2 1 0
D. The program could produce thousands of lines of output.
E. The program could deadlock before producing any output.
F. The output could be "5 ", followed by the program deadlocking.

Question 3.
What will the following program print?

```
public class TestClass{
	static boolean b;
	static int[] ia = new int[1];
	static char ch;
	static boolean[] ba = new boolean[1];
	public static void main(String args[]) throws Exception{
		boolean x = false;
		if( b ){
			x = ( ch == ia[ch]);
		}
		else x = ( ba[ch] = b );
		System.out.println(x+" "+ba[ch]);
	}
}
```

Select 1 option:
A. true true
B. true false
C. false true
D. false false
E. It will not compile.

Question 4.
Given:

```
import java.io.*;
public class Edgy {
	public static void main(String[] args) {
		try {
			wow();
			// throw new IOException(); // line 6
		} finally {
			// throw new Error(); // line 8
			// throw new IOException(); // line 9
		}
	}

	static void wow() {
		// throw new IllegalArgumentException(); //line 13
		// throw new IOException(); // line 14
	}
}
```

And given that `IOException` is a direct subclass of `java.lang.Exception`: and that `IllegalArgumentException` is a runtime exception, which of the following, if uncommented independently, will compile?

Choose all that apply:
A. Line 6
B. Line 8
C. Line 9
D. Line 13
E. Line 14
F. The code will NOT compile as is.

Question 5.
Given two files:

```
package com;
public class MyClass {
	public static void howdy() { System.out.print("howdy "); }
	public static final int myConstant = 343;
	public static final MyClass mc = new MyClass();
	public int instVar = 42;
}
```

```
import com.MyClass;
public class TestImports2 {
	public static void main(String[] args) {
		MyClass.howdy(); // line 14
		System.out.print(MyClass.myConstant + " "); // line 15
		System.out.print(myConstant + " "); // line 16
		howdy(); // line 17
		System.out.print(mc.instVar + " "); // line 18
		System.out.print(instVar + " "); // line 19
	}
}
```

What is the result?

Choose all that apply:
A. howdy 343 343 howdy 42 42
B. Compilation fails due to an error on line 14.
C. Compilation fails due to an error on line 15.
D. Compilation fails due to an error on line 16.
E. Compilation fails due to an error on line 17.
F. Compilation fails due to an error on line 18.
G. Compilation fails due to an error on line 19.

Question 6.
Given this code inside a method:

```
int count = 0;
outer: for(int x = 0; x < 5; x++) {
	middle: for(int y = 0; y < 5; y++) {
		if(y == 1) continue middle;
		if(y == 3) break middle;
		count++;
	}

	if(x > 2) continue outer;
	count = count + 10;
}

System.out.println("count: " + count);
```

What is the result?

Select 1 option:
A. count: 33
B. count: 40
C. count: 45
D. count: 65
E. Compilation fails.
F. The code runs in an endless loop.

Question 7.
Given:

```
public class Organic<E> {
       void react(E e) { }
       static void main(String[] args) {
              // Organic<? extends Organic> compound = new Aliphatic<Organic>(); // line 5
              // Organic<? super Aliphatic> compound = new Aliphatic<Organic>(); // line 6
              compound.react(new Organic()); // line 7
              compound.react(new Aliphatic()); //line 8
              compound.react(new Hexane()); // line 9
       }
}

class Aliphatic<F> extends Organic<F> { }
class Hexane<G> extends Aliphatic<G> { }
```

Which, taken independently, are true?

Choose all that apply:
A. If line 5 is uncommented, compilation fails due to an error at line 7.
B. If line 5 is uncommented, compilation fails due to an error at line 8.
C. If line 5 is uncommented, compilation fails due to an error at line 9.
D. If line 6 is uncommented, compilation fails due to an error at line 7.
E. If line 6 is uncommented, compilation fails due to an error at line 8.
F. If line 6 is uncommented, compilation fails due to an error at line 9.

Question 8.
Which of these statements about interfaces are true?

Choose all that apply:
A. Interfaces are always abstract.
B. An interface can have static methods.
C. All methods in an interface are abstract although you need not declare them to be so.
D. Fields of an interface may be declared as transient or volatile but not synchronized.
E. Interfaces cannot be final.
F. In Java 8, interfaces allow multiple implementation inheritance through default methods.

Question 9.
What will be a part of the output when the following code is compiled and run?

```
class Boo {
	int boo = 10;
	public Boo(int k){ System.out.println("In Boo k = "+k); boo = k;}
}

class BooBoo extends Boo {
	public BooBoo(int k ){ super(k); System.out.println("In BooBoo k = "+k); }
}

class Moo extends BooBoo implements Serializable {
	int moo = 10;
	public Moo(){ super(5); System.out.println("In Moo"); }
}

public class TestClass {

	public static void main(String[] args) throws Exception{
		Moo moo = new Moo();
		FileOutputStream fos = new FileOutputStream("c:\\temp\\moo1.ser");
		ObjectOutputStream os = new ObjectOutputStream(fos);
		os.writeObject(moo);
		os.close();

		FileInputStream fis = new FileInputStream("c:\\temp\\moo1.ser");
		ObjectInputStream is = new ObjectInputStream(fis);
		moo = (Moo) is.readObject();
		is.close();
	}
}
```

Choose all that apply:

A. `In Boo k = 5`
`In BooBoo k = 5`
B. `In Boo k = 0`
`In BooBoo k = 0`
C. `In Moo`
D. It will throw an exception at runtime.
E. It will not compile.

Question 10.
Given:

```
public static void main(String[] args) {
	try {
		throw new Error(); // line 5
	}
	catch (Error e) { // line 7
		try { throw new RuntimeException(); } // line 8
		catch (Throwable t) { } // line 9
	}

	System.out.println("phew");
}
```

Which are true?

Choose all that apply:

A. The output is phew
B. The code runs without output.
C. Compilation fails due to an error on line 5.
D. Compilation fails due to an error on line 7.
E. Compilation fails due to an error on line 8.
F. Compilation fails due to an error on line 9.

Question 11.
Given:

```
import java.util.*;
public class Bucket {
	public static void main(String[] args) {
		Set<String> hs = new HashSet<String>();
		Set<String> lh = new LinkedHashSet<String>();
		Set<String> ts = new TreeSet<String>();
		List<String> al = new ArrayList<String>();
		String[] v = {"1", "3", "1", "2"};

		for(int i=0; i< v.length; i++) {
			hs.add(v[i]); lh.add(v[i]); ts.add(v[i]); al.add(v[i]);
		}

		Iterator it = hs.iterator();

		while(it.hasNext()) System.out.print(it.next() + " ");

		Iterator it2 = lh.iterator();

		while(it2.hasNext()) System.out.print(it2.next() + " ");

		Iterator it3 = ts.iterator();

		while(it3.hasNext()) System.out.print(it3.next() + " ");

		Iterator it5 = al.iterator(); // line 18

		while(it5.hasNext()) System.out.print(it5.next() + " ");
	}
}
```

Which statements are true?

Choose all that apply:
A. An exception is thrown at runtime.
B. Compilation fails due to an error on line 18.
C. "1 3 2" is only guaranteed to be in the output once.
D. "1 2 3" is only guaranteed to be in the output once.
E. "1 3 2" is guaranteed to be in the output more than once.
F. "1 2 3" is guaranteed to be in the output more than once.
G. "1 3 1 2" is guaranteed to be in the output at least once.
H. Compilation fails due to error(s) on lines other than line 18.

Question 12.
Given:

```
public class Tshirt extends Thread {
	public static void main(String[] args) {
		System.out.print(Thread.currentThread().getId() + " "); // line 4
		Thread t1 = new Thread(new Tshirt());
		Thread t2 = new Thread(new Tshirt());
		t1.start();
		t2.run(); // line 8
	}

	public void run() {
		for(int i = 0; i < 2; i++)
		System.out.print(Thread.currentThread().getId() + " ");
	}
}
```

Which are true?

Choose all that apply:

A. No output is produced.
B. The output could be 1 1 9 9 1
C. The output could be 1 2 9 9 2
D. The output could be 1 9 9 9 9
E. An exception is thrown at runtime.
F. Compilation fails due to an error on line 4.
G. Compilation fails due to an error on line 8.

Question 13.
Consider the following code in which `searchBook()` method has an inner class.

```
public class BookStore {
       private static final int taxId = 300000;
       private String name;
       public String searchBook(final String criteria) {
              int count = 0;
              int sum = 0;
              sum++;
              class Enumerator {
                     String interate( int k) {
                            //line 1
                            // lots of code
                            return "";
                     }

                     // lots of code.....
              }

              // lots of code.....
              return "";
       }
}
```

If the code will compile, which variables are accessible at line 1?

Choose all that apply:

A. `taxId`
B. `name`
C. `criteria`
D. `count`
E. `k`
F. sum
G. Compilation fails

Question 14.
Given:

```
public class Clover extends Harrier {
       String bark() { return "feed me "; }
       public static void main(String[] args) {
              Dog[] dogs = new Dog[3];
              dogs[0] = new Harrier(); // line 6
              dogs[1] = (Dog)new Clover(); // line 7
              dogs[2] = (Dog)new Harrier(); // line 8

              for(Dog d: dogs) System.out.print(d.bark()); // line 9
       }
}

class Dog { String bark() { return "bark "; } }
class Harrier extends Dog { String bark() { return "woof "; } }
```

What is the result?

Choose all that apply:

A. bark bark bark
B. woof bark bark
C. woof feed me woof
D. Compilation fails due to an error on line 6.
E. Compilation fails due to an error on line 7.
F. Compilation fails due to an error on line 8.
G. Compilation fails due to an error on line 9.

Question 15.
Consider:

```
public class Outer
{
  int i = 10;
  class Inner
  {
    public void methodA()
    {
       //line 1.
    }
  }
}
```

Which of the following statements are valid at line 1.

Select 2 options:

A. `System.out.println(this.i);`
B. `System.out.println(i);`
C. `System.out.println(Outer.this.i);`
D. '`i`' cannot be accessed inside the inner class method.
E. The code cannot be compiled.

Question 16.
Carefully examine the following code.

```
public class StaticTest {
       static {
              System.out.println("In static");
       }

       {
              System.out.println("In non - static");
       }

       public static void main(String args[ ]) {
              StaticTest st1;                    // line 1
              System.out.println(" 1 ");
              st1 = new StaticTest();            // line 2
              System.out.println(" 2 ");
              StaticTest st2 = new StaticTest(); // line 3
       }
}
```

What will be the output?

Select 1 option:

A. `In static, 1, In non - static, 2, In non - static` :in that order.
B. Compilation error.
C. `1, In static, In non - static, 2, In non - static` : in that order.
D. `In static, 1, In non - static, 2, In non - static` : in unknown order.

Question 17.
What will the following code print?

```
class Test{
	public static void main(String[] args){
		int k = 1;
		int[] a = { 1 };
		k += (k = 4) * (k + 2);
		a[0] += (a[0] = 4) * (a[0] + 2);
		System.out.println( k + " , " + a[0]);
	}
}
```

Select 1 option:
A. It will not compile.
B. 4,4
C. 25,25
D. 13,13
E. None of the above.

Question 18.
Given:

```
class Jog implements Runnable {
	public void run() {
		for(int i = 0; i < 8; i++) {
			try { Thread.sleep(200); }
			catch (Exception e) { System.out.print("exc "); }
			System.out.print(i + " ");
		}
	}
}

public class Marathon {
	public static void main(String[] args) throws Exception {
		Jog j1 = new Jog();
		Thread t1 = new Thread(j1);
		t1.start();
		t1.sleep(500);
		System.out.print("pre ");
		t1.interrupt();
		t1.sleep(500);
		System.out.print("post ");
	}
}
```

Assuming that `sleep()` sleeps for about the amount of time specified in its argument, and that all other code runs almost instantly, which output is likely?

Choose all that apply:
A. exc
B. 0 1 pre exc post
C. exc 0 1 2 3 4 5 6 7
D. pre post 0 1 2 3 4 5 6 7
E. pre exc 0 1 post 2 3 4 5 6 7
F. 0 1 pre exc 2 3 4 post 5 6 7

Question 19.
Consider the following code:

```
import java.util.List;
import java.util.concurrent.CopyOnWriteArrayList;
public class MyCache {
	private CopyOnWriteArrayList<String> cal = new CopyOnWriteArrayList<>();

	public void addData(List<String> list){
		cal.addAll(list);
	}

	public Iterator getIterator(){
		return cal.iterator();
	}
}
```

Given that one thread calls the `addData` method on an instance of the above class and another thread calls the `getIterator` method on the same instance at the same time and starts iterating through its values, which of the following options are correct?
(Assume that no other calls have been made on the MyCache instance.)

Select 1 option:
A. The call to `addAll` may be blocked while the other thread is iterating through the iterator.
B. Both the threads will complete their operations successfully without getting any exception.
C. The thread iterating through the Iterator will get a `ConcurrentModificationException`.
D. Elements added by the `addAll` method will automatically be visible through the iterator in the other thread.

Question 20.
Consider the following code:

```
public class Student {
	private Map<String, Integer> marksObtained = new HashMap<String, Integer>();
	private ReadWriteLock lock = new ReentrantReadWriteLock();

	public void setMarksInSubject(String subject, Integer marks){
		// 1 INSERT CODE HERE
		try{
			marksObtained.put(subject, marks);
		}finally{
			// 2 INSERT CODE HERE
		}

	}

	public double getAverageMarks(){
		// valid code that computes average
	}
}
```

What should be inserted at //1 and //2?

Select 1 option:
A. lock.writeLock().acquire(); and lock.writeLock().release();
B. lock.writeLock().lock(); and lock.writeLock().unlock();
C. lock.acquire(); and lock.release();
D. lock.readLock().lock(); and lock.readLock().unlock();

Question 21.
Given:

```
public class ItemProcessor implements Runnable { //LINE 1
	CyclicBarrier cb;
	public ItemProcessor(CyclicBarrier cb){
		this.cb = cb;
	}

	public void run(){
		System.out.println("processed");
		try {
			cb.await();
		} catch (Exception ex) {
			ex.printStackTrace();
		}
	}
}

public class Merger implements Runnable { //LINE 2
	public void run(){
		System.out.println("Value Merged");
	}
}
```

What should be inserted in the following code such that run method of `Merger` will be executed only after the thread started at 4 and the main thread have both invoked await?

```
	public static void main(String[] args) throws Exception{
		Merger m = new Merger();

		//LINE 3

		ItemProcessor ip = new ItemProcessor(cb);
		ip.start(); //LINE 4
		cb.await();
	}
```

Select 1 option:

A. Make ItemProcessor extend Thread instead of implementing Runnable and add CyclicBarrier cb = new CyclicBarrier(1, m); to //LINE 3
B. Make ItemProcessor extend Thread instead of implementing Runnable and add CyclicBarrier cb = new CyclicBarrier(2, m); to //LINE 3
C. Make Merger extend Thread instead of implementing Runnable and add CyclicBarrier cb = new CyclicBarrier(1, m); to //LINE 3
D. Make Merger extend Thread instead of implementing Runnable and add CyclicBarrier cb = new CyclicBarrier(2, m); to //LINE 3
E. Just add CyclicBarrier cb = new CyclicBarrier(1, m); to //LINE 3
F. Just add CyclicBarrier cb = new CyclicBarrier(2, m); to //LINE 3

Question 22.
Given:

```
public class Counter{
	//1
	public  void increment(){
		//2
	}

	//other valid code
}
```

This class is supposed to keep an accurate count for the number of times the increment method is called. Several classes share an instance of this class and call its increment method. What should be inserted at //1 and //2?

Select 1 option:

A. int count; at //1 and AtomicInteger.increment(count); at //2
B. synchronized int count; at //1 and count++ at //2
C. AtomicInteger count = 0; at //1 and count++; at //2
D. AtomicInteger count = new AtomicInteger(0); at //1 and count++; at //2
E. AtomicInteger count = new AtomicInteger(0); at //1 and count.incrementAndGet(); at //2

Question 23.
Consider the following code:

```
public class Test extends Thread {
	boolean flag = false;
	public Test(boolean f) { flag = f; }
	static Object obj1 = new Object();
	static Object obj2 = new Object();

	public void m1() {
		synchronized(obj1) {
			System.out.print("1 ");
			synchronized(obj2) {
				System.out.println("2");
			}
		}
	}

	public void m2() {
		synchronized(obj2) {
			System.out.print("2 ");
			synchronized(obj1) {
				System.out.println("1");
			}
		}
	}

	public void run() {
		if(flag){ m1();  m2(); }
		else { m2();  m1(); }
	}

	public static void main(String[] args) {
		new Test(true).start();
		new Test(false).start();
	}
}
```

Which of the following statements are correct?

Select 2 options:

A. It may result in a deadlock and the program may get stuck.
B. There is no potential for deadlock in the code.
C. Deadlock may occur but the program will not get stuck as the JVM will resolve the deadlock.
D. The program will always print: 1 2, 2 1, 1 2 and 2 1.
E. Nothing can be said for sure.

Question 24.
Which of the following statements are true?

Select 1 option:

A. The Thread class does not implement Runnable.
B. The Thread class is abstract.
C. Thread is an interface which has only one method, namely public void run();
D. Calling the method run( ) on an object castable to Thread will start the new thread. (Assuming that run() method is not overridden.)
E. Calling the method start( ) on an object castable to Thread will start the new thread. (Assuming that start() method is not overridden.)

Question 25.
Which of the following statements about this program are correct?

```
class CoolThread extends Thread {
	String id = "";
	public CoolThread(String s){ this.id = s; }

	public void run() {
		System.out.println(id+" End");
	}

	public static void main(String args []) {
		Thread t1 = new CoolThread("AAA");
		t1.setPriority(Thread.MIN_PRIORITY);
		Thread t2 = new CoolThread("BBB");
		t2.setPriority(Thread.MAX_PRIORITY);
		t1.start(); t2.start();
	}
}
```

Select 1 option:
A. "AAA End" will always be printed before "BBB End".
B. "BBB End" will always be printed before "AAA End".
C. The order of "AAA End" and "BBB End" cannot be determined.
D. The program will not compile.
E. The program will throw an exception at runtime.

Question 26.
Which statements regarding the following program are correct?

```
class A extends Thread {
	static protected int i = 0;

	public void run() {
		for(; i<5; i++) System.out.println("Hello");
	}
}

public class TestClass extends A {
	public void run() {
		for(; i<5; i++)   System.out.println("World");
	}

	public static void main(String args []) {
		Thread t1 = new A();
		Thread t2 = new TestClass();
		t2.start(); t1.start();
	}
}
```

Select 1 option:
A. It will not compile because run method cannot be overridden.
B. It will print both "Hello" and "World" 5 times each.
C. It will print both "Hello" and "World" 5 times each but they may be interspersed.
D. Total 5 words will be printed.
E. Either 5 "Hello"s or 5 "world"s will be printed.
F. None of these.

Question 27.
What will the following code print when run?

```
Deque<Integer> d = new ArrayDeque<>();
d.push(1);
d.push(2);
d.push(3);
System.out.println(d.remove());
System.out.println(d.remove());
System.out.println(d.remove());
```

Select 1 option:

A. 1
2
3

B. 3
2
1

C. Compilation error

D. Exception at run time

Question 28.
Given the following complete contents of `TestClass.java`:

```
import java.util.ArrayList;
import java.util.Collections;

class Address implements Comparable<Address> {
	String street;
	String zip;
	public Address(String street, String zip){
		this.street = street; this.zip = zip;
	}

	public int compareTo(Address o) {
		int x = this.zip.compareTo(o.zip);
		return x == 0? this.street.compareTo(o.street) : x;
	}
}

public class TestClass {
	public static void main(String[] args) {
		ArrayList<Address> al = new ArrayList<>();
		al.add(new Address("dupont dr", "28217"));
		al.add(new Address("sharview cir", "28217"));
		al.add(new Address("yorkmont ridge ln", "11223"));
		Collections.sort(al);

		for(Address a : al) System.out.println(a.street+" "+ a.zip);
	}
}
```

What will be printed?

Select 1 option:

A. yorkmont ridge ln 11223
dupont dr 28217
sharview cir 28217

B. sharview cir 28217
yorkmont ridge ln 11223
dupont dr 28217

C. sharview cir 28217
dupont dr 28217
yorkmont ridge ln 11223

D. The order of output messages cannot be determined.

E. It will throw an exception at run time.

F. It will not compile.

Question 29.
Which of these statements concerning maps are true?

Select 1 option:
A. It is permissible for a map to contain itself as a key.
B. values() method returns an instance of Set.
C. The Map interface extends the Collection interface.
D. All keys in a map are unique.
E. All Map implementations keep the keys sorted.

Question 30.
What will be printed when the following code is compiled and run?

```
package trywithresources;

import java.io.IOException;

public class Device implements AutoCloseable {
	String header = null;

	public void open(){
		header = "OPENED";
		System.out.println("Device Opened");
	}

	public String read() throws IOException{
		throw new IOException("Unknown");
	}

	public void writeHeader(String str) throws IOException{
		System.out.println("Writing : "+str);
		header = str;
	}

	public void close(){
		header = null;
		System.out.println("Device closed");
	}

	public static void testDevice(){
		try(Device d = new Device()){
			d.open();
			d.read();
			d.writeHeader("TEST");
			d.close();
		}catch(IOException e){
			System.out.println("Got Exception");
		}
	}

	public static void main(String[] args) {
		Device.testDevice();
	}
}
```

Select 1 option:
A. Device Opened
Device closed
Got Exception
B. Device Opened
Got Exception
Device closed
C. Device Opened
Got Exception
D. The code will not compile.

Question 31.
Consider the following code snippet :

```
public void readData(String fileName) throws Exception {
      try(FileReader fr1 = new FileReader(fileName)){
            Connection c = getConnection();
            //...other valid code
      }
}
```

Given that `getConnection` method throws `ClassNotFoundException` and that an `IOException` is thrown while closing `fr1`, which exception will be received by the caller of `readData()` method?

Select 1 option:
A. java.lang.Exception
B. java.lang.ClassNotFoundException
C. java.io.IOException
D. java.lang.RuntimeException

Question 32.
Given:

```
Map<String , List<? extends CharSequence>> stateCitiesMap =
new HashMap<String, List<? extends CharSequence>>();
```

Which of the following options correctly achieves the same declaration using type inferencing?

Select 1 option:
A. Map<String , List<? extends CharSequence>> stateCitiesMap = new HashMap<String, List<>>();
B. Map<String , List<? extends CharSequence>> stateCitiesMap = new HashMap();
C. Map<String , List<? extends CharSequence>> stateCitiesMap = new HashMap<>();
D. Map<String , List<? extends CharSequence>> stateCitiesMap = new HashMap<<>>();

Question 33.
Given:

```
class Game{ }
class Cricket extends Game{ }
class Instrument{ }
class Guitar extends Instrument{ }

interface Player<E>{ void play(E e); }
interface GamePlayer<E extends Game> extends Player<E>{ }
interface MusicPlayer<E extends Instrument> extends Player{ }
```

Identify valid declarations.

Select 1 option:
A.
```
class Batsman implements GamePlayer<Cricket>{
    public void play(Game o){ }
}
```
B.
```
class Bowler implements GamePlayer<Guitar>{
    public void play(Guitar o){ }
}
```
C.
```
class Bowler implements Player<Guitar>{
    public void play(Guitar o){ }
}
```
D.
```
class MidiPlayer implements MusicPlayer {
    public void play(Guitar g){ }
}
```
E.
```
class MidiPlayer implements MusicPlayer<Instrument> {
    public void play(Guitar g){ }
}
```

Question 34.
Given:

```
String[] p = {"1", "2", "3" };
```

Which of the following lines of code is valid?

Select 1 option:
A. List<?> list2 = new ArrayList<Integer>(Arrays.asList(p));
B. List<Integer> list2 = new ArrayList<Integer>(Arrays.asList(p));
C. List<Integer> list2 = new ArrayList<>(Arrays.asList(p));
D. List<?> list2 = new ArrayList<>(Arrays.asList(p));

Question 35.
Given the following class:

```
public class SpecialPicker<K> {
	public K pickOne(K k1, K k2){
		return k1.hashCode()>k2.hashCode()?k1:k2;
	}
}
```

what will be the result of running the following lines of code:

```
	SpecialPicker<Integer> sp = new SpecialPicker<>(); // line 1
	System.out.println(sp.pickOne(1, 2).intValue()+1); // line 2
```

Select 1 option:
A. line 1 will not compile.
B. line 2 will not compile.
C. It will throw an exception at run time.
D. It will print 3.

Question 36.
Replace XXX with a declaration such that the following code will compile without any error or warning.

```
	public void m1(XXX list)
	{
	Number n = list.get(0);
	}
```

Select 1 option:
A. List<? super Number>
B. List<?>
C. List<? extends Number>
D. List<Number extends ?>
E. List<Number super ?>

Question 37.
Given:

```
public class PlaceHolder<K, V> { //1
	private K k;
	private V v;
	public PlaceHolder(K k, V v){ //2
		this.k = k;
		this.v = v;
	}

	public K getK() { return k; }
	public static <X> PlaceHolder<X, X> getDuplicateHolder(X x){ //3
		return new PlaceHolder<X, X>(x, x); //4
	}
}
```

Which line will cause compilation failure?

Select 1 option:
A. 1
B. 2
C. 3
D. 4
E. Some other unnumbered line.
F. There is no problem with the code.

Question 38.
The signature of a method in a class is as follows:

```
public static <E extends CharSequence> List<? super E> doIt(List<E> nums)
```

This method is being called in the following code:

```
result = doIt(in);
```

Given that `String` implements `CharSequence` interface, what should be the reference type of `'in'` and `'result'` variables?

Select 1 option:
A.
```
ArrayList<String> in;
List<CharSequence> result;
```
B.
```
List<String> in;
List<Object> result;
```
C.
```
ArrayList<String> in;
List result;
```
D.
```
List<CharSequence> in;
List<CharSequence> result;
```
E.
```
ArrayList<Object> in;
List<CharSequence> result;
```
F. None of these.

Question 39.
Which statements regarding the following code are correct?

```
class A extends Thread {
	public void run() {
		System.out.println("Starting loop");
		while(true){}
	}
}

public class TestClass {
	public static void main(String args[]) throws Exception {
		A a = new A();
		a.start();
		Thread.sleep(1000);
		a.interrupt();
	}
}
```

Select 1 option:
A. This will not compile.
B. It will compile but it will throw an exception at Runtime.
C. It will run but it will never end.
D. It will run and will end after about a second.
F. None of these.

Question 40.
Given the following code:

```
public class Data implements java.io.Serializable {
	public static String f1;
	public static transient int f2;
	public transient boolean f3;
	public final static String f4 = "4";
	public String f5 = "5";
}

Data d = new Data();
d.f1 = "f1";
d.f2 = 2;
d.f3 = true;
```

Which fields of the above class will be preserved when the object referred to by the variable `'d'` is serialized and deserialized in another JVM?

Select 1 option:
A. f1, f4, f5
B. f4, f5
C. f5
D. f3, f5