```java
package exam2;
import java.util.LinkedList;
import java.util.List;

class A             {}
class B extends X {}
class C extends B   {}
class D extends C   {}

public class Q1<T extends X> {

    T instance;
    public Q1(T instance) {
        this.instance = instance;
    }

    public static void main(String[] args)  {
        List<Q1<? super B>> cList = new LinkedList<Q1<? super B>>();
//        cList.add( new Q1<A>(new A() ) );
//        cList.add( new Q1<B>(new B() ) );
//        cList.add( new Q1<C>(new C() ) );
//        cList.add( new Q1<D>(new D() ) );
    }
}

// Explain in English what the first commented line would do if
uncommented

// For all 4 commented lines, which ones will prevent compilation?
// Explain why for each line

```

```
 1 package exam2;
 2 import java.util.ArrayList;
 3 import java.util.Collection;
 4 public class Q2<T> {
 5     public static void main(String[] args)  {
 6         Collection c = new ArrayList<Q1>();
 7         c.add(new Q1(new X()));
 8         c.add(new Q1(new B()));
 9 //         c.add(new Q1(new Object()));
10     }
11 }
12
13 // For this question, consider first question's source code as part of
    the same package
14 // If this code is compiled with -Xlint, will any message be printed?
   If so, describe the meaning of message. Assume all dependencies
   successfully compile.
15 // If the commented line is uncommented, will this code compile? Why?
16
```

```
 1 package exam2;
 2
 3 public class Q3 {
 4
 5     public void run() {
 6         int[] intArray = {2};
 7         try {
 8             try {
 9                 try {
10                     System.out.println("" + intArray[1]);
11                     System.out.println("Exception 0 ");
12                 } catch(Exception e) {
13                     System.out.println("Exception 1 ");
14                 }
15             } catch(Exception e){
16                 System.out.println("" + intArray[1]);
17                 System.out.println("Exception 2 ");
18             }
19         } catch(Exception e) {
20             System.out.println("Exception 3 ");
21         } finally {
22             System.out.println("finally 2 ");
23         }
24     }
25     public static void main(String[] args) { new Q3().run(); }
26 }
27
28 // What will be the output of this program?
29 // For all lines that are not executed, explain the reason for that.
30
```

```
 1 package exam2;
 2 import java.io.*;
 3
 4 public class Q4 implements Serializable {
 5     transient Integer value = 42;
 6
 7     static void read() {
 8         try (ObjectInputStream in = new ObjectInputStream(new
   FileInputStream("object.save"));) {
 9             Q4 o = (Q4) in.readObject();
10 //            o.value = 42;
11             System.out.println("Value = " + o.value);
12         } catch (Exception e) { System.out.println(e.getMessage()); }
13     }
14
15     static void write() {
16         try (ObjectOutputStream out = new ObjectOutputStream(new
   FileOutputStream("object.save"));) {
17             Q4 o = new Q4();
18             out.writeObject(o);
19         } catch (Exception e) { System.out.println(e.getMessage()); }
20     }
21
22     public static void main (String args[] ){
23         write();
24         read();
25     }
26 }
27
28 // What is the output of this program?
29 // Is the object fully serialized? Why?
30 // If the commented line is uncommented, what would the output of the
   program be?
```

```java
package exam2;

abstract class AbstractClass<T> {
    abstract void go(T aT);
}

interface Interface<T> {
//    abstract void go(T aT);
}

class X<T> implements Interface<T> {
    void go(T aT)   { System.out.print("running go()"); }
    public void go() {}
}

class Y<T> extends AbstractClass<T> {
    void go(T aT)   {}
    public static void main(String args[])  {
        AbstractClass<Integer> aX = new Y<Integer>();
    }
}

// This program compiles as is.
// What is the effective access modifier used for the abstract method go() in AbstractClass?
// What is the effective access modifier used for the abstract method go() in Interface?
// If the commented line is uncommented, the program will not compile. Why?
// After uncommenting the commented line, what modification in class X would
//    allow the program to compile? Explain your solution.


```