Chapter 11 – Exceptions

Question 1:

Given the following class hierarchy and the code fragments below

```
                 java.lang.Exception
                        \
                  java.io.IOException
                   /                 \
          FirstException        SecondException
```

```
 1.try
 2.{
 3.    Car car = new Car();
 4.    Object o = car.getEngine();
 5.    System.out.println("One");
 6.}
 7.catch (SecondException e)
 8.{
 9.    System.out.println("Two");
10.}
11.catch (FirstException e)
12.{
```

```
13.     System.out.println("First");
14.}
15. catch (Exception e)
16.{
17.     System.out.println("General");
18.}
19.finally
20.{
21.     System.out.println("finally");
22.}
23.System.out.println("keep on going");
```

What will be included in the output if the method at line 4 throws a FirstException ?

[] A) First

[] B) Second

[] C) General

[] D) finally

[] E) keep on going

Question 2:

Which one of the following options shows the most appropriate way to throw an exception ?

```
()    A)    Exception ex = new SQLException(“Data wasn’t found”);
            if (!f.exist())
            {
                  throw ex;
             }
()    B)    if (!f.exists())
            {
                       throw new SQLException(“Data wasn’t found”);
               }
()    C)    if (!f.exists())
            {
                       throw new SQLException;
               }
()    D)    if (!f.exists())
            {
                       throw “Data wasn’t found”;
               }
()    E)    if (!f.exists())
            {
                       throw SQLException;
            }
```

Question 3:

Consider the following class hierarchy and code fragments:

```
            java.lang.Throwable
                 /                \
     java.lang.Error      java.lang.Exception
              /                             \
          AError                       AException
            /                                    \
        BError                                BException
```

```
1. try
2. {
3.      Car auto=new Car();
4.      Object o = auto.readEngine();
5.      System.out.println("Success");
6. }
7. catch (Bexception e)
8. {
9.      System.out.println("Bexception has occured");
10.}
11.catch (Aexception e)
12.{
13.     System.out.println("Aexception has occured");
14.}
15.catch (Exception e)
16.{
17.     System.out.println("General exception has occured");
18.}
19.finally
20.{
21.     System.out.println("doing finally part");
22.}
23.System.out.println("Carrying on");
```

What lines are output if the method at line 4 throws a Berror ?

[]   A)   Bexception has occurred

[]   B)   Aexception has occurred

[]   C)   General Exception has occurred

[]   D)   doing finally part

[]   E)   Carrying on

Question 4:

Consider the following class hierarchy and code fragments:

```
            java.lang.Throwable
              /                    \
     java.lang.Error      java.lang.Exception
              /                          \
         AError                    AException
           /                                   \
        BError                           BException
```

```
1. try
2. {
3.      Car auto=new Car();
4.      Object o = auto.readEngine();
5.      System.out.println("Success");
6. }
7. catch (Bexception e)
8. {
9.      System.out.println("Bexception has occured");
10.}
11.catch (Aexception e)
12.{
13.     System.out.println("Aexception has occured");
}
catch (Exception e)
{
    System.out.println("General exception has occured");
}
finally
{
    System.out.println("doing finally part");
}
System.out.println("Carrying on");
```

What lines are output if the try block works without throwing any Exception ?

[] A) Bexception has occurred

[] B) Aexception has occurred

[] C) General Exception has occurred

[] D) doing finally part

[] E) Carrying on

[] F) Success

Question 5:

Which of the following statements are true ?

[] A) An overriding method may legitimately throw fewer exceptions than its original, but it may not throw more.

[] B) An overriding method can be less accessible than the original method

[] C) An overriding method cannot be less accessible than the original method

[] D) An overriding method must be able to throw exactly the same Exceptions as the original one.

[] E) An overriding method must have the same accessible permission as the original.

Question 6:

Given the following class hierarchy and the code fragments below

```
            Java.lang.Exception
                      \
           java.lang.BambaException
                         \
               java.io.IOException
                 /                  \
      FirstException           SecondException
```

```
 1.try
 2.{
 3.    Car car = new Car();
 4.    Object o = car.getEngine();
 5.    System.out.println("One");
 6.}
 7.catch (SecondException e)
 8.{
 9.    System.out.println("Two");
10.}
11.catch (FirstException e)
12.{
13.    System.out.println("First");
14.}
15. catch (IOException e)
16.{
17.    System.out.println("General");
18.}
19.finally
20.{
21.    System.out.println("finally");
22.}
23.System.out.println("keep on going");
```

What will be included in the output if the method at line 4 throws a BambaException ?

[] A) First

[] B) Second

[] C) General

[] D) finally

[] E) keep on going

Question 7:

Given the following class hierarchy and the code fragments below

```
            Java.lang.Exception
                      \
           java.lang.BambaException
                        \
               java.io.IOException
                /               \
      FirstException        SecondException
```

```
1. class Base
2. {
3.      public int getSize(int num) throws IOException {}
4. }
```

```
1. class Derive extends Base
```

```
2. {
3.
4. }
```

What are the methods that can be declared in class Derive (in line 3) ?

[] A) public float getSize(float num) throws BambaException {}

[] B) public float getSize(int num) throws IOException {}

[] C) public int getSize(int num1) throws IOException {}

[] D) public int getSize(int num1) throws Exception {}

[] E) public int getSize(int num1) throws SecondException {}

Question 8:

Given the code below,

```
1.    public class MyApplet extends Applet
2.    {
3.            public void init()
4.            {
5.                    String str = “Israel”;
6.                    try
7.                    {
8.                            str = getParameter(“SIZE”).toUpperCase();
```

```
        }
        catch(Exception e)
        {
            str = "Tel-Aviv";
        }
    }
}

<applet code="MyApplet.class" width=100 height=60>
<param name="Size" value="Israel2000">
</applet>
```

[] A) The value of str is “Tel-Aviv”

[] B) The value of str is “ISRAEL2000”

[] C) The value of str is “israel2000”

[] D) The value of str is “Israel2000”

[] E) The value of str is null

[] F) The try & catch block isn’t necessary

Question 9:

Under what circumstances the finally block might be stopped or might not be activated ?

[] A) The thread in which the finally block is executed will be stopped
as a result of activating the stop method from that thread.

[] B) An exception will be thrown during the execution of the finally block.

[] C) The exit method in the System class will be activated.

[] D) If an exception will be caught by a catch statement which is placed
before the finally block then the finally block won't run.

[] E) If the code that relates to the finally block runs without throwing any exception then the
finally block won't be activated.

Question 10:

Given the following class hierarchy and the code fragments below

```
                 Java.lang.RuntimeException
                           \
                java.lang.BambaException
                             \
                     java.io.IOException
                     /                 \
            FirstException          SecondException
```

```
 1.try
 2.{
 3.    Car car = new Car();
 4.    Object o = car.getEngine();
 5.    System.out.println("One");
 6.}
 7.catch (SecondException e)
 8.{
 9.    System.out.println("Two");
10.}
11.catch (FirstException e)
12.{
13.    System.out.println("First");
14.}
15. catch (IOException e)
16.{
17.    System.out.println("General");
18.}
19.finally
20.{
21.    System.out.println("finally");
22.}
23.System.out.println("keep on going");
```

What will be included in the output if the method at line 4 throws a BambaException ?

[] A) First

[] B) Second

[] C) General

[] D) finally

[] E) keep on going

Question 11:

True or False:

An applet can’t write to files in the computer in which it is executed.

() A) True

() B) False

Question 12:

Given the following class hierarchy and the code fragments below

```
                 java.lang.Exception
                          \
                  java.io.IOException
                  /                  \
          FirstException        SecondException
```

```
 1.try
 2.{
 3.    Car car = new Car();
 4.    Object o = car.getEngine();
 5.    System.out.println("One");
 6.}
 7.catch (SecondException e)
 8.{
 9.    System.out.println("Two");
10.}
11.catch (FirstException e)
12.{
13.    System.out.println("First");
14.}
15. catch (Exception e)
```

```
16.{
17.     System.out.println("General");
18.}
19.finally
20.{
21.     System.out.println("finally");
22.}
23.System.out.println("keep on going");
```

What will be included in the output if the method at line 4 throws a IOException ?

[] A) First

[] B) Second

[] C) General

[] D) finally

[] E) keep on going

Question 13:

Given the following class hierarchy and the code fragments below

```
                 java.lang.Exception
                         \
                  java.io.IOException
                    /               \
          FirstException        SecondException
```

```
 1.try
 2.{
 3.    Car car = new Car();
 4.    Object o = car.getEngine();
 5.    System.out.println("One");
 6.}
 7.catch (SecondException e)
 8.{
 9.    System.out.println("Two");
10.}
11.catch (FirstException e)
12.{
13.    System.out.println("First");
14.}
15.finally
```

```
16.{
17.     System.out.println("finally");
18.}
19.System.out.println("keep on going");
```

What will be included in the output if the method at line 4 throws a IOException ?

[] A) First

[] B) Second

[] C) General

[] D) finally

[] E) keep on going

Question 14:

Given the following class hierarchy and the code fragments below

```
                    java.lang.Exception
                          \
                     java.io.IOException
                      /                  \
              FirstException          SecondException
               /                        \
      ThirdException                FourthException
```

```
 1.try
 2.{
 3.    Car car = new Car();
 4.    Object o = car.getEngine();
 5.    System.out.println("One");
 6.}
 7.catch (SecondException e)
 8.{
 9.    System.out.println("Two");
10.}
11.catch (FirstException e)
12.{
13.    System.out.println("First");
14.}
15.finally
16.{
17.    System.out.println("finally");
18.}
19.System.out.println("keep on going");
```

What will be included in the output if the method at line 4 throws a FourthException ?

[] A) First

[] B) Second

[] C) General

[] D) finally

[] E) keep on going

[] F) Two

Question 15:

Given the following class hierarchy and the code fragments below

```
                  java.lang.Exception
                          \
                    java.io.IOException
                     /               \
          FirstException        SecondException
```

```
try
{
    Car car = new Car();
    Object o = car.getEngine();
    System.out.println("One");
}
catch (SecondException e)
{
    System.out.println("Two");
}
```

```
11.catch (FirstException e)
12.{
13.     System.out.println("First");
14.}
15.finally
16.{
17.     System.out.println("finally");
18.}
19.System.out.println("keep on going");
```

What will be included in the output if the method at line 4 throws a FirstException ?

[]   A)   First

[]   B)   Second

[]   C)   General

[]   D)   finally

[]   E)   keep on going

Question 16:

Given the following class hierarchy and the code fragments below

```
                  java.lang.Exception
                         \
                   java.io.IOException
                    /               \
          FirstException        SecondException
```

```
 1.try
 2.{
 3.    Car car = new Car();
 4.    Object o = car.getEngine();
 5.    System.out.println("One");
 6.}
 7.catch (SecondException e)
 8.{
 9.    System.out.println("Two");
10.}
11.catch (Exception e)
12.{
13.    System.out.println("First");
14.}
15.finally
```

```
16.{
17.     System.out.println("finally");
18.}
19.System.out.println("keep on going");
```

What will be included in the output if the method at line 4 throws a SQLException ?

[]   A)   First

[]   B)   Second

[]   C)   General

[]   D)   finally

[]   E)   keep on going

<u>Question 17:</u>

Given the following class hierarchy and the code fragments below

```
                 Java.lang.Exception
                        \
                java.lang.BambaException
                          \
                   java.io.IOException
                    /                 \
          FirstException         SecondException
```

```
1. class Base
2. {
3.      public int getSize(int num) throws IOException {}
4. }
```

```
1. class Derive extends Base
2. {
3.
4. }
```

What are the methods that can be declared in class Derive (in line 3) ?

[] A) public double getSize() throws BambaException {}

[] B) public float getSize() {}

[] C) public int getSize(int num) throws IOException {}

[] D) public int getSize(int num) throws SecondException {}

[] E) public int getSize(int num, int value) throws Exception {}

Question 18:

True or False:

Exceptions are thrown when semantic constraints are violated and cause a non-local transfer of control from the point where the exceptions occurred to a point that can be specified by the programmer. The exception is thrown from the point where it occurred and caught at the point to which the control is transferred.

[] A) True

[] B) False

Question 19:

True or False:

The finally block is optional, and it allows you to have code that will always run, whether or not an error occurs.

[] A) True

[] B) False

Question 20:

True or False:

The RuntimeException classes constitute a special case: They do not have to be declared or caught. All other exception objects require a conscious effort on your part, which means the following: You can either catch them or explicitly declare that you might throw them. If needed, you can do both.

[] A) True

[] B) False

Question 21:

True or False:

Instances of the Error class (The Error class extends the Throwable class) represent abnormal conditions that should not occur (such as errors in the JVM). Recovering from them is usually not feasible. Since it is impossible to recover from errors there’s not much point in catching them. If you do have a chance of recovering from a specific error, you should catch it.

[] A) True

[] B) False

Question 22:

True or False:

The given code runs infinitely.

```
import java.util.*;
public class ServerOfArrays
{
        private Hashtable bankOfArrays = new Hashtable();
        public Object getArray(String key)
        {
                Object result = bankOfArrays.get(key);
                if(result==null)
                {
                        try
                        {
                                result = new long[1000000];
                                bankOfArrays.put(key,result);
                                System.out.println("new array was instantiated and was added” +
                                                                        “to the hash table");
                        }
                        catch(OutOfMemoryError error)
                        {
                                System.out.println("OUT OF MEMORY");
                                bankOfArrays.clear();
                System.out.println("MEMORY WAS CLEANED");
                result = getArray(key);
            }
        }
        return result;
    }
    public static void main(String args[])
    {
        ServerOfArrays endlessArrays = new ServerOfArrays();
        int counter=0;
        Object ob = null;
        while(true)
        {
            ob = endlessArrays.getArray(""+counter);
            System.out.println("vec num."+counter+" was recieved\t");
            counter++;
        }
    }
}
```

[] A) True

[] B) False

Question 23:

True or False:

The OutOfMemoryError is an Error type from which it might be possible to recover.

[] A) True

[] B) False