Module5-I/O 流

一、选择题

Question 1
Given:

```
10. class MakeFile {
11. public static void main(String[] args) {
12. try {
13. File directory = new File("d");
14. File file = new File(directory,"f");
15. if(!file.exists()) {
16. file.createNewFile();
17. }
18. } catch (IOException e) {
19. e.printStackTrace();
20. }
21. }
22. }
```

The current directory does NOT contain a directory named "d."
Which three are true? (Choose three.)

A. Line 16 is never executed.
B. An exception is thrown at runtime.
C. Line 13 creates a File object named "d."
D. Line 14 creates a File object named "f."
E. Line 13 creates a directory named "d" in the file system.
F. Line 16 creates a directory named "d" and a file 'f' within it in the file system.
G. Line 14 creates a file named "f" inside of the directory named "d" in the file system.

Answer: BCD

Question 2
When comparing java.io.BufferedWriter to java.io.FileWriter, which capability exists as a method in only one of the two?
A. closing the stream
B. flushing the stream
C. writing to the stream
D. marking a location in the stream
E. writing a line separator to the stream

Answer: E

Copyright Tarena Corporation,2009.All rights reserved

Question 3
Which three concerning the use of the java.io.Serializable interface are true? (Choose three.)
A. Objects from classes that use aggregation cannot be serialized.
B. Art object serialized on one JVM can be successfully deserialized on a different JVM.
C. The values in fields with the volatile modifier will NOT survive serialization and deserialization.
D. The values in fields with the transient modifier will NOT survive serialization and deserialization.
E. It is legal to serialize an object of a type that has a supertype that does NOT implement java.io.Serializable.

Answer: BDE

Question 4
Assuming that the serializeBanana() and the deserializeBanana() methods will correctly use Java serialization and given:

```
13. import java.io.*;
14. class Food implements Serializable {int good = 3;}
15. class Fruit extends Food {int juice = 5;}
16. public class Banana extends Fruit {
17. int yellow = 4;
18. public static void main(String [] args) {
19. Banana b = new Banana(); Banana b2 = new Banana();
20. b.serializeBanana(b);                    // assume correct serialization
21. b2 = b.deserializeBanana();              // assume correct
22. System.out.println("restore "+b2.yellow+ b2.juice+b2.good);
24. }
25. // more Banana methods go here
50. }
```

‘What is the result?
A. restore 400
B. restore 403
C. restore 453
D. Compilation fails.
E. An exception is thrown at runtime.

Answer: C

Question 5
Assuming that the serializeBanana2() and the deserializeBanana2() methods will correctly use Java serialization and given:

```
13. import java.io.*;
14. class Food {Food() { System.out.print("1"); } }
15. class Fruit extends Food implements Serializable {
```

```
16. Fruit() { System.out.print("2"); } }
17. public class Banana2 extends Fruit { int size = 42;
18. public static void main(String [] args) {
19. Banana2 b = new Banana2();
20. b.serializeBanana2(b);                    // assume correct serialization
21. b = b.deserializeBanana2(b);              // assume correct
22. System.out.println(" restored "+ b.size + " "); }
23. // more Banana2 methods
24. }
```

What is the result?

A. Compilation fails.
B. 1 restored 42
C. 12 restored 42
D. 121 restored 42
E. 1212 restored 42
F. An exception is thrown at runtime.

Answer: D

Question 6
Given:

```
10. public class Foo implements java.io.Serializable {
11. private int x;
12. public int getX() { return x; }
12.publicFoo(int x){this.x=x; }
13. private void writeObject( ObjectOutputStream s)
14. throws IOException {
15. // insert code here
16. }
17. }
```

Which code fragment, inserted at line 15, will allow Foo objects to be correctly serialized and deserialized?

A. s.writeInt(x);
B. s.serialize(x);
C. s.writeObject(x);
D. s.defaultWriteObject();

Answer: D

Question 7
Click the Exhibit button.

```
1. import java.io.*;
2. public class Foo implements Serializable {
3. public int x, y;
4. public Foo( int x, int y) { this.x = x; this.y = y; }
5.
6. private void writeObject( ObjectOutputStream s)
7. throws IOException {
8. s.writeInt(x); s.writeInt(y)
9. }
10.
11. private void readObject( ObjectInputStream s)
12. throws IOException, ClassNotFoundException {
13.
14. // insert code here
15.
16. }
17. }
```

Which code, inserted at line 14, will allow this class to correctly serialize and deserialize?

A. s.defaultReadObject();
B. this = s.defaultReadObject();
C. y = s.readInt(); x = s.readInt();
D. x = s.readInt(); y = s.readInt();

Answer: D

Question 8

Given:

```
12. import java.io.*;
13. public class Forest implements Serializable {
14. private Tree tree = new Tree();
15. public static void main(String [] args) {
16. Forest f= new Forest();
17. try {
18. FileOutputStream fs = new FileOutputStream("Forest.ser");
19. ObjectOutputStream os = new ObjectOutputStream(fs);
20. os.writeObject(f); os.close();
21. } catch (Exception ex) { ex.printStackTrace(); }
22. } }
23.
24. class Tree { }
```

What is the result?

A. Compilation fails.
B. An exception is thrown at runtime.
C. An instance of Forest is serialized.
D. A instance of Forest and an instance of Tree are both serialized.

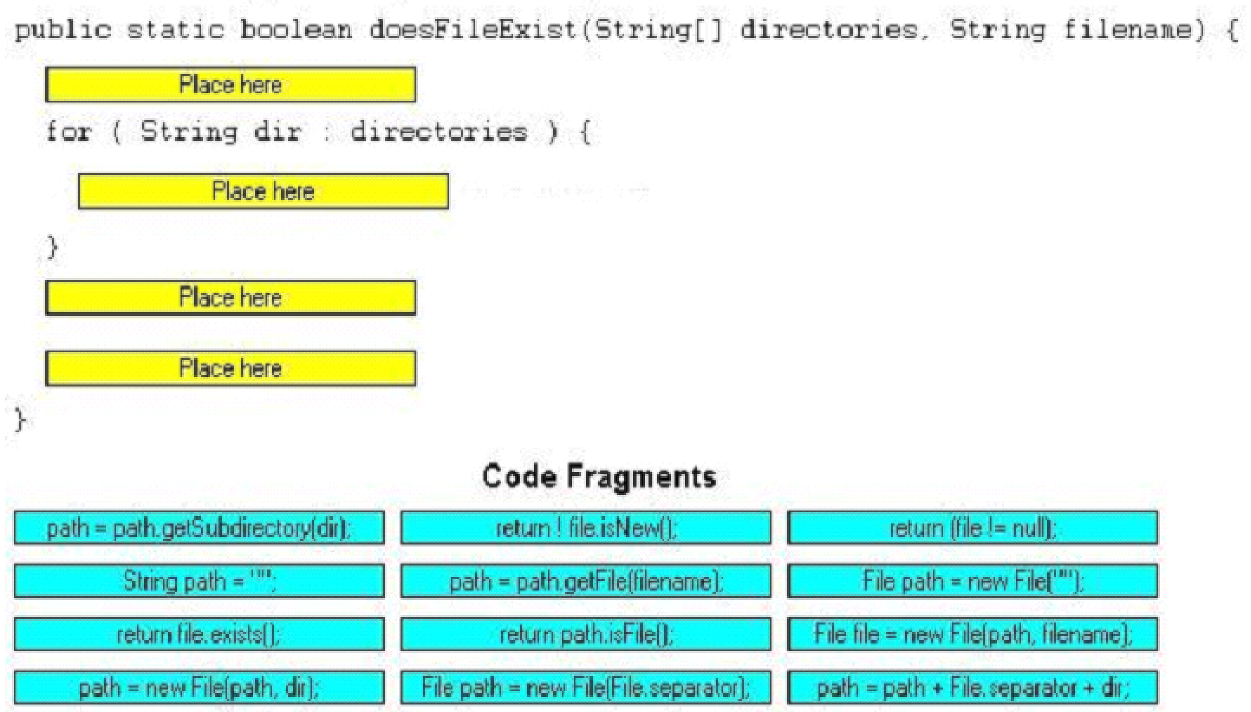
Answer: B

Copyright Tarena Corporation,2009.All rights reserved

二、拖拽题：

Question 1:

The doesFileExist method takes an array of directory names representing a path from the root filesystem and a file name. The method returns true if the file exists, false if does not. Place the code fragments in position to complete this method.



```
public static boolean doesFileExist(String[] directories, String filename) {
    Place here
    for ( String dir : directories ) {
        Place here
    }
    Place here

    Place here
}
```

Code Fragments

path = path.getSubdirectory(dir);	return ! file.isNew();	return (file != null);
String path = "";	path = path.getFile(filename);	File path = new File("");
return file.exists();	return path.isFile();	File file = new File(path, filename);
path = new File(path, dir);	File path = new File(File.separator);	path = path + File.separator + dir;

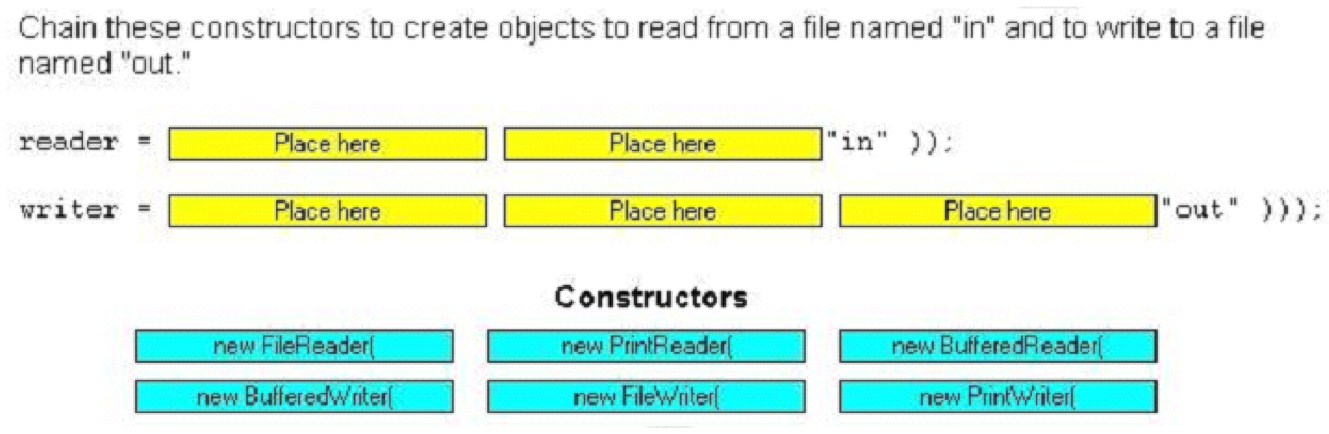
Answer:

```
String path = "";
for(String dir : directories)
{
    path = path + File.separator + dir;
}

File file = new File(path, filename);
return file.exists();
```

Copyright Tarena Corporation,2009.All rights reserved

Question 2:



Chain these constructors to create objects to read from a file named "in" and to write to a file named "out."

```
reader = [Place here] [Place here]"in" ));

writer = [Place here] [Place here] [Place here]"out" )));
```

Constructors

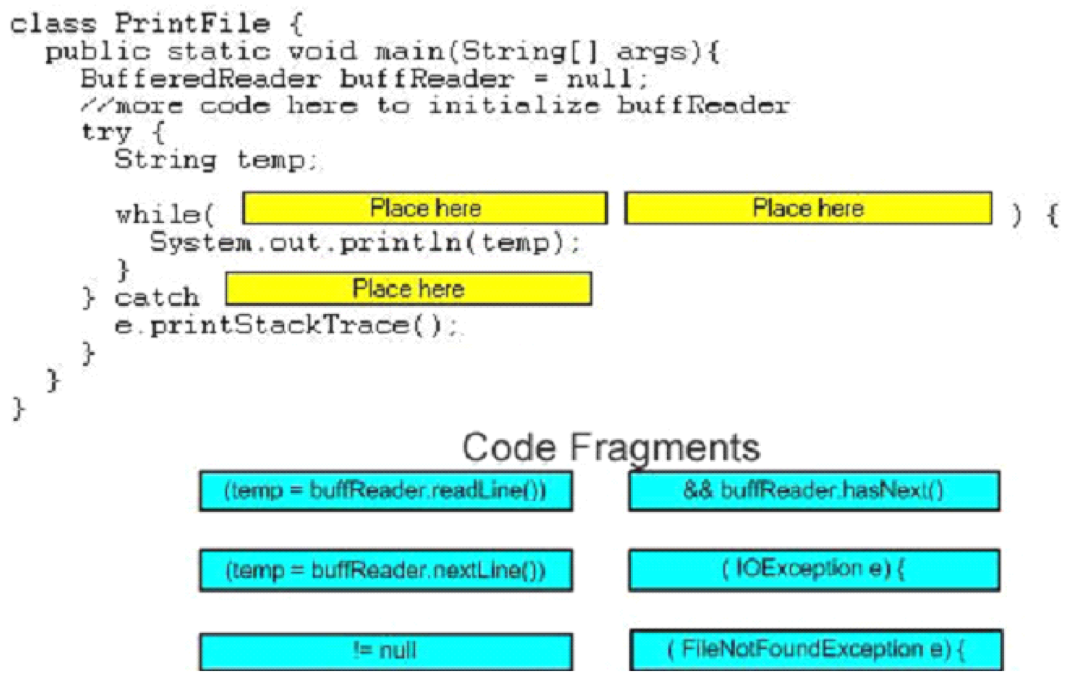
new FileReader(	new PrintReader(	new BufferedReader(
new BufferedWriter(	new FileWriter(	new PrintWriter(

Answer: reader=new BufferedReader(new FileReader("in"));
writer=new BufferedWriter(new PrintWriter(new FileWriter("out")));
或 writer=new PrintWriter(new BufferedWriter(new FileWriter("out")));

Copyright Tarena Corporation,2009.All rights reserved

Question 3:

Place the code fragments into position to use a BufferedReader to read in an entire text file.



```
class PrintFile {
  public static void main(String[] args){
    BufferedReader buffReader = null;
    //more code here to initialize buffReader
    try {
      String temp;

      while( [Place here] [Place here] ) {
        System.out.println(temp);
      }
    } catch [Place here]
      e.printStackTrace();
    }
  }
}
```

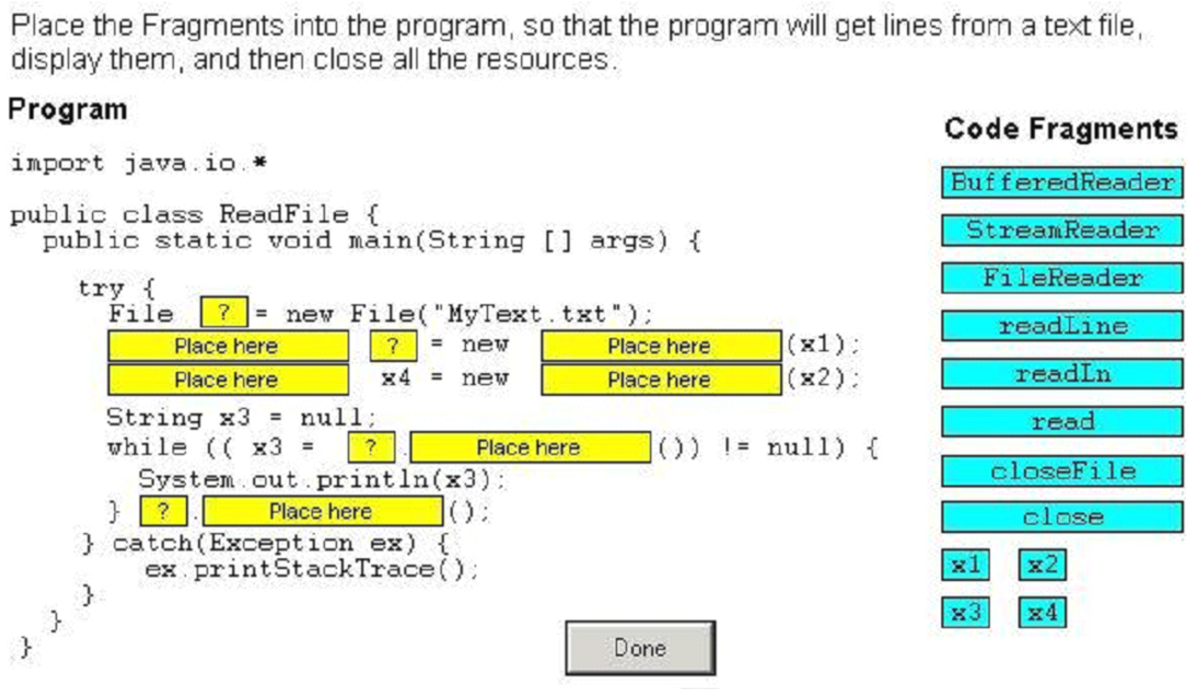
Code Fragments

```
(temp = buffReader.readLine())     && buffReader.hasNext()
(temp = buffReader.nextLine())     ( IOException e) {
!= null                            ( FileNotFoundException e) {
```

Answer:

```
try{
        String temp;
        while((temp=buffReader.readLine())!=null){
                System.out.println(temp);
        }
}catch(IOException e){
        e.printStackTrace();
}
```

Question 4:



Place the Fragments into the program, so that the program will get lines from a text file, display them, and then close all the resources.

Program

```
import java.io.*

public class ReadFile {
  public static void main(String [] args) {

    try {
      File [?] = new File("MyText.txt");
      [Place here] [?] = new [Place here](x1);
      [Place here]  x4 = new [Place here](x2);
      String x3 = null;
      while (( x3 = [?].[Place here]()) != null) {
        System.out.println(x3);
      } [?].[Place here]();
    } catch(Exception ex) {
       ex.printStackTrace();
    }
  }
}
```

Code Fragments

BufferedReader
StreamReader
FileReader
readLine
readLn
read
closeFile
close
x1 x2
x3 x4

Answer:

```
import java.io.*;
public class ReadFile {
	public static void main(String[] args) {
		try{
			File x1 = new File("MyText.txt");
			feredReader x4 = new BufferedReader(x2);
			String x3 = null;
			while((x3 = x4.readLine()) != null){
				System.out.println(x3);
			}
			x4.close();
		}catch(Exception ex){
			ex.printStackTrace();
		}
	}
}
```