

1. Which is the package you import to work with files?

- a. java.lang
- b. java.util
- c. java.io
- d. java.awt

2. Which is the super class for Byte stream classes?

- a. StringBufferInputStream
- b. DataInput
- c. Object
- d. InputStream

3. What is the exception thrown if the control reaches the end of file?

- a. FileNotFoundException
- b. EOFException
- c. IOException
- d. Exception

4. Analyse this code snippet :

```
import java.io.FileReader;
import java.io.FileWriter;
import java.util.Scanner;
public class SystemInfo {
    public static void main(String[] args) {
        try {
            FileWriter fw = new FileWriter("one.txt");
            fw.write("Kiran,Mahesh,Rajesh");
            fw.close();
            FileReader fr = new FileReader("one.txt");
            Scanner scan = new Scanner(fr);
            while (scan.hasNext()) {
                System.out.println(scan.next() + " ");
            }
        }
    }
}
```

```

        }
    } catch (Exception e) {
        System.out.println("Exception -1 ");
    }
}

```

} What is the output?

- A. Kiran
Mahesh
Rajesh
- B. Kiran,Mahesh,Rajesh
- C. Compilation Error
- D. Compilation successful but it throws RuntimeException

5. Analyse this code snippet :

```

package com.spameos.collection;
import java.io.DataInputStream;
import java.io.IOException;
public class SystemInfo {
    public static void main(String[] args) {
        DataInputStream fi = new DataInputStream(System.in);
        try{
        }catch(IOException e){
            System.exit(0);
        }
        finally {System.out.println("Doing finally");}
    }
}

```

What would be the result when you compile and run this program?

- A. Compilation Error
- B. Compilation successful but no output

- C. Doing finally
- D. None of the above

6. The file "file.txt" exists on the file system and contains ASCII text. Given:

```
try {  
    File f = new File("file.txt");  
    OutputStream out = new FileOutputStream(f, true);  
} catch (IOException e) {}
```

 What is the result?

- A. The code does not compile.
- B. The code runs and no change is made to the file.
- C. The code runs and sets the length of the file to 0.
- D. An exception is thrown because the file is not closed.

7. Analyse this code snippet:

```
package com.spameos.collection;  
  
import java.io.File;  
  
public class SystemInfo {  
    public static void main(String[] args) {  
        File dir = new File("dir");  
        File f = new File(dir, "f");  
    }  
}
```

 What is the result?

- A. Compilation fails.
- B. Nothing is added to the file system.
- C. Only a new file is created on the file system.
- D. Only a new directory is created on the file system.

8. Analyse this code snippet:

```
package com.spameos.collection;  
  
import java.io.FileOutputStream;  
import java.io.ObjectOutputStream;  
  
class Person{
```

```

    private String name;
    public Person(String name) {
        this.name=name;
    }
    @Override
    public String toString() {
        return this.name;
    }
}

public class SystemInfo {
    public static void main(String[] args) {
        try{
            System.out.println(" Main method");
            FileOutputStream fis=new FileOutputStream("one.txt");
            Person p=new Person("Core java");
            ObjectOutputStream ois=new ObjectOutputStream(fis);
            ois.writeObject(p);
            ois.close();

        }catch(Exception e){
            System.out.println(e);
        }
    }
}

```

} What happens when you compile and run this program?

- A. Compilation Error
- B. Compilation successful but no output
- C. Main method
- D. Runtime Exception: java.io.NotSerializableException

9. Analyse this code snippet:

```
import java.io.Serializable;
```

```

class Person implements Serializable{
    private static String name;
    public Person(String name) {
        this.name=name;
    }
    @Override
    public String toString() {
        return this.name;
    }
}

public class SystemInfo {
    public static void main(String[] args) {
        try{
            FileOutputStream fis=new FileOutputStream("one.txt");
            Person p=new Person("Core java");
            ObjectOutputStream ois=new ObjectOutputStream(fis);
            ois.writeObject(p);
            ois.writeObject(new Person("Servlets"));
            ois.close();

        }catch(Exception e){
            System.out.println(e);
        }
    }
}

```

} What happens when you compile and run this program?

- A. Compilation Error
- B. Compilation Successful but at run time, message from JVM is “static variable can't be serialized”
- C. Compilation Successful and program executes but no output
- D. Runtime Exception: java.io.NotSerializableException

10. Fill the blanks to deserialize Person object from the one.txt file

```
try{
    FileInputStream fis=new FileInputStream("one.txt");
    ObjectInputStream ois=_____
    Person person=(_____)ois._____;
    ois.close();

}catch(Exception e){
    System.out.println(e);
}
```

- A. new ObjectInputStream(fis)
 Person
 readPerson()
- B. new ObjectInputStream(fis)
 Person
 readObject()
- C. new ObjectOutputStream(fis)
 Person
 readPerson()
- D. None of the above

11. Analyze this code snippet :

```
package com.spameos.collection;
import java.io.FileOutputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
class Car{}
class Person implements Serializable{
    public Car car=new Car();
    public void writeIn(){
        try{
            FileOutputStream fos=new FileOutputStream("one.txt");
```

```

        ObjectOutputStream oos=new ObjectOutputStream(fos);
        oos.writeObject(car);
        System.out.println("Done");
        oos.close();
    }catch(Exception e){
        System.out.println("Exception-1");
    }
}
}

public class SystemInfo {
    public static void main(String[] args) {
        Person person=new Person();
        person.writeIn();
    }
}

```

What is the output?

- A. Done
- B. Done
Exception-1
- C. Exception-1
- D. None of the above

12. Analyze this code snippet :

```

package com.spaneos.collection;
import java.io.*;
class Person_c implements Serializable{}
class Employee_c extends Person_c {
}

public class ObjectStoring {
    public static void main(String[] args) {
        try{
            FileOutputStream fos=new FileOutputStream("one.txt");

```

```

        ObjectOutputStream oos=new ObjectOutputStream(fos);
        Employee_c emp=new Employee_c();
        oos.writeObject(emp);
        System.out.println("Done");
        oos.close();
    }catch(Exception e){
        System.out.println("Exception-1");
    }
}

```

} What is the output?

- A. Compilation Error
- B. Exception -1
- C. Done
- D. Runtime Excpetion Employee_c not implemented Serilizable Interface

13. Analyze this code snippet:

```

package com.spaneos.collection;
import java.io.*;
public class ObjectStoring {
    public static void main(String[] args) {
        try{
            File file=new File("one.txt");
            file._____;
            FileReader fr=new _____ (file);
            int ch;
            while((ch=fr._____)!=-1)
                System.out.println((char)ch);
        }catch(Exception e){
            System.out.println("Exception-1");
        }
    }
}

```


} Fill in the blanks to compile and run this program.

A. createNewFile()

 FileReader

 read()

B. newFile()

 FileReader

 read

C. createNewFile()

 FileWriter

 write

D. None of the above

14. Analyze this code snippet:

```
package com.spaneos.collection;
import java.io.*;
public class ObjectStoring {
    public static void main(String[] args) {
        try{
            File file=new File("c:/");
            if(file._____){
                File f[]=file._____;
                for(_____ fi:f){
                    System.out.println(fi);
                }
            }
        }catch(Exception e){
            System.out.println("Exception-1");
        }
    }
}
```

} Fill in the blanks to compile and run this program successfully (windows)

A. directory()

listFiles()

File

B. isDirectory()

listFiles()

File

C. isDirectory()

filesList()

File

D. None of the above

15. Analyze this code snippet:

```
package com.spameos.collection;
```

```
import java.io.*;
```

```
public class ObjectStoring {
```

```
    public static void main(String[] args) {
```

```
        try{
```

```
            File file=new File("Two.txt");
```

```
            file.createNewFile();
```

```
            System.out.println("file is created ");
```

```
        }catch(Exception e){
```

```
            System.out.println("Exception-1"+file);
```

```
        }
```

```
    }
```

```
} What happens when you compile and run this program?
```

A. Compilation Error

B. file is created

C. Exception -1 file

D. Runtime Exception FileNotFoundException

16. Fill in the blanks to create a file in the specified directory

```
public class Example {  
    public static void main(String[] args) {  
        try{  
            File directory=new File("myfiles");  
            directory._____ ();  
            File f=new File(_____, "one.txt");  
            f._____ ();  
        }catch(_____ e){  
            e.printStackTrace();  
        }  
    }  
}
```

- A. Mkdir directory createNewFile FileNotFoundException
- B. Mkdir directory createFile Exception
- C. CreateFolder directory createNewFile Exception
- D. Mkdir directory createNewFile Exception

17. What is the output of the following code snippet?

```
class One implements Serializable{ }  
  
class Two extends One{ }  
  
public class Example {  
    public static void main(String[] args) {  
        try{  
            FileOutputStream fos=new FileOutputStream("one.txt");  
            ObjectOutputStream oos=new ObjectOutputStream(fos);  
            oos.writeObject(new Two());  
            System.out.println("Writing object successful");  
        }catch(Exception e){  
            e.printStackTrace();  
        }  
    }  
}
```

```

    }
}
}

```

- A. Class Two can't be serialized because It has not implemented Serializable interface
- B. Program compiles and runs but no output
- C. Program compiles successfully and prints "Writing object successful"
- D. Program leads to run time exception : java.io.FileNotFoundException

18. What is the output of the following code snippet?

```

class One implements Serializable{
    public One() {        System.out.print("One");        }
}
public String toString() {        return "three";        }

}

class Two extends One{
    Two(){        System.out.print(" Two");        }
    public String toString() {        return " four";        }
}

public class Example {
    public static void main(String[] args) {
        try{
            FileOutputStream fos=new FileOutputStream("one.txt");
            ObjectOutputStream oos=new ObjectOutputStream(fos);
            oos.writeObject(new Two());
            FileInputStream fis=new FileInputStream("one.txt");
            ObjectInputStream ois=new ObjectInputStream(fis);
            Two two=(Two) ois.readObject();
            System.out.print(" "+two);
        }catch(Exception e){
            e.printStackTrace();
        }
    }
}

```

```
    }  
}
```

- A. Four
- B. One Two four One Two
- C. One Two
- D. One Two One Two Four
- E. One Two four

19. What is the output of the following code snippet?

```
public class Example {  
    public static void main(String[] args) {  
        try{  
            OutputStream out = new FileOutputStream("one.txt");  
            out.write(22); out.close();  
            InputStream input = new FileInputStream("one.txt");  
            System.out.println(input.read() + " " + input.read());  
            input.close();  
        }catch(Exception e){  
            e.printStackTrace();  
        }  
    }  
}
```

- A. 22 and Exception
- B. 22 -1
- C. 22 0
- D. 22 -0
- E. 22 1

20. What is the output of the following code snippet?

```
public class Example {  
    public static void main(String[] args) {  
        try{  
            1.    String name = null;  
            2.    File file = new File(name);  
            3.    System.out.println(file.getName());  
        }catch(Exception e){  
            e.printStackTrace();  
        }  
    }  
}
```

- A. Line number 2 NullPointerException
- B. Line number 3 NullPointerException
- C. Line number 1 NullPointerException
- D. It creates new File
- E. Creates new file with anonymous name