

Java Software Development Exercise 4

1. Which one of these methods in the `File` class will return the name of the entry, excluding the specification of the directory in which it resides? Select the one correct answer.

- (A) `getAbsolutePath()`
- (B) `getName()`
- (C) `getParent()`
- (D) `getPath()`
- (E) None of the above.

2. Given the following program:

```
import java.io.File;
import java.io.IOException;
public final class Filing {
    public static void main (String[] args) throws IOException {
        File file = new File("./documents", "../book/../chapter1");
        System.out.println(file.getPath());
        System.out.println(file.getAbsolutePath());
        System.out.println(file.getCanonicalPath());
        System.out.println(file.getName());
        System.out.println(file.getParent());
    }
}
```

Assume that the current or working directory has the absolute path `"/wrk"`. Which lines below will not be included in the output from the program? Select the two correct answers.

- (A) `./documents/../book/../chapter1`
- (B) `./documents/book/chapter1`
- (C) `/wrk/../documents/../book/../chapter1`
- (D) `/wrk/documents/book/chapter1`
- (E) `/wrk/chapter1`
- (F) `chapter1`
- (G) `./documents/../book/..`

3. Which of these classes provides methods for writing binary representations of Java primitive values? Select the two correct answers.

- (A) `DataOutputStream`
- (B) `FileOutputStream`
- (C) `ObjectOutputStream`
- (D) `PrintStream`
- (E) `BufferedOutputStream`

4. Which of the following best describes the data written by an `ObjectOutputStream`? Select the one correct answer.

- (A) Bytes and other Java primitive types.
- (B) Object hierarchies.
- (C) Object hierarchies and Java primitive types.
- (D) Single objects.
- (E) Single objects and Java primitive types.

5. Given the following code:

```
public class Person {
    protected String name;
```

```

    Person() { }
    Person(String name) { this.name = name; }
}

import java.io.Serializable;
public class Student extends Person implements Serializable {
    private long studNum;
    Student(String name, long studNum) {
        super(name);
        this.studNum = studNum;
    }
    public String toString() { return "(" + name + ", " + studNum + ")"; }
}

import java.io.*;

public class RQ800_10 {
    public static void main(String args[]) throws IOException,
        ClassNotFoundException {
        FileOutputStream outputFile = new FileOutputStream("storage.dat");
        ObjectOutputStream outputStream=new ObjectOutputStream(outputFile);
        Student stud1 = new Student("Aesop", 100);
        System.out.print(stud1);
        outputStream.writeObject(stud1);
        outputStream.flush();
        outputStream.close();
        FileInputStream inputFile = new FileInputStream("storage.dat");
        ObjectInputStream inputStream = new ObjectInputStream(inputFile);
        Student stud2 = (Student) inputStream.readObject();
        System.out.println(stud2);
        inputStream.close();
    }
}

```

Which statement about the program is true? Select the one correct answer.

- (A) It fails to compile.
- (B) It compiles, but throws an exception at runtime.
- (C) It prints (Aesop, 100)(Aesop, 100).
- (D) It prints (Aesop, 100)(null, 100).
- (E) It prints (Aesop, 100)(, 100).

Answer

1. (B)

The method `getName()` can be used on a `File` object to return the name of the entry excluding the specification of the directory in which the entry resides.

2. (B) and (D)

Compiling and running the program results in the following output:

```
./documents/../book/../chapter1  
/wrk/../documents/../book/../chapter1  
/wrk/chapter1  
chapter1  
./documents/../book/..
```

3. (A) and (C)

Classes that implement the `DataOutput` interface, i.e., `DataOutputStream` and `ObjectOutputStream`, provide methods for writing binary representations of primitive values. The output stream classes `FileOutputStream`, `PrintStream`, and `BufferedOutputStream` do not provide such methods.

4. (C)

A `ObjectOutputStream` can write both objects and Java primitive types, as it implements the `ObjectInput` and the `DataInput` interfaces. The serialization mechanism will follow object references and can write whole hierarchies of objects.

5. (D)

During deserialization, the default constructor of the superclass `Person` is called, because this superclass is not `Serializable`.