

I/o Lab1

The image displays an IDE with five Java files and an output window. The files are:

- Circle.java**:

```
1 package Lab1;
2 public class Circle implements Shape{
3
4     @Override
5     public void draw() {
6         System.out.println("Shape: Circle");
7     }
8
9 }
10
```
- ShapeDecorator.java**:

```
1 package Lab1;
2
3 public abstract class ShapeDecorator implements Shape{
4     protected Shape shape;
5     public ShapeDecorator(Shape shape) {
6         this.shape = shape;
7     }
8
9     @Override
10    public void draw() {
11        shape.draw();
12    }
13
14 }
```
- RedShape.java**:

```
1 package Lab1;
2
3 public class RedShape extends ShapeDecorator{
4     public RedShape(Shape shape) {
5         super(shape);
6     }
7
8     @Override
9     public void draw() {
10        shape.draw();
11        System.out.println("Border Color: Red");
12    }
13
14 }
```
- Rectangle.java**:

```
1 package Lab1;
2
3 public class Rectangle implements Shape{
4
5     @Override
6     public void draw() {
7         System.out.println("Shape: Rectangle");
8     }
9
10 }
11
```
- DecoratorTest.java**:

```
1 package Lab1;
2
3 public class DecoratorTest {
4     public static void main(String[] args) {
5         Circle obj = new Circle();
6         Rectangle obj2 = new Rectangle();
7
8         RedShape redShape = new RedShape(obj);
9         RedShape redShape2 = new RedShape(new Rectangle());
10        redShape.draw();
11        redShape2.draw();
12    }
13
14 }
```

The **Output** window shows the following text:

```
run:
Shape: Circle
Border Color: Red
Shape: Rectangle
Border Color: Red
BUILD SUCCESSFUL (total time: 0 seconds)
```

Lab2

The screenshot shows an IDE with a Java file named `Lab2.java`. The code defines a `Lab2` class with a `main` method. The `main` method first writes the string "Hello World" to a file named `Lab2.txt` using `FileOutputStream`. It then reads the file back using `FileInputStream` and prints each character on a new line using `System.out.println`. The output window shows the result of the program execution, displaying "Hello World" on separate lines. The build was successful.

```
1 package Lab2;
2
3 import java.io.*;
4 import java.io.IOException;
5 public class Lab2 {
6     public static void main(String[] args) throws IOException {
7         FileOutputStream out = new FileOutputStream("Lab2.txt");
8         String s = "Hello World";
9         byte[] b = s.getBytes();
10        out.write(b);
11        out.close();
12
13        FileInputStream in = new FileInputStream("Lab2.txt");
14        int value;
15        while ((value = in.read()) != -1) {
16            System.out.println((char) value);
17        }
18    }
19 }
```

Output:

```
run:
H
e
l
l
o

W
o
r
l
d
BUILD SUCCESSFUL (total time: 0 seconds)
```

Lab3

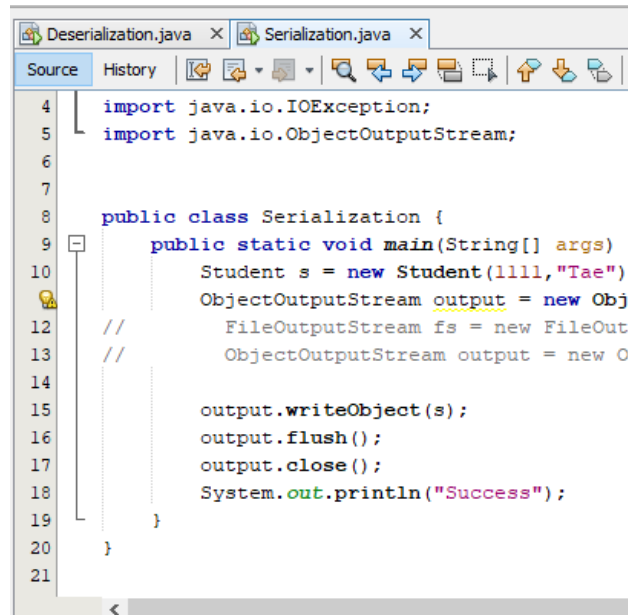
The screenshot shows an IDE with two panels. The top panel, titled 'Lab3.java', contains the following Java code:

```
1 package Lab3;
2 import java.io.*;
3 public class Lab3 {
4     public static void main(String[] args) throws FileNotFoundException, IOException {
5         DataOutputStream output = new DataOutputStream(new FileOutputStream("C:\\Users\\TUA\\Desktop\\demo.txt"));
6         output.writeInt(12352);
7         output.writeDouble(25.5);
8         output.writeChar('A');
9         output.writeUTF("Jim");
10        output.flush();
11        output.close();
12
13        DataInputStream input = new DataInputStream(new FileInputStream("C:\\Users\\TUA\\Desktop\\demo.txt"));
14        System.out.println(input.readInt());
15        System.out.println(input.readDouble());
16        System.out.println(input.readChar());
17        System.out.println(input.readUTF());
18        input.close();
19    }
}
```

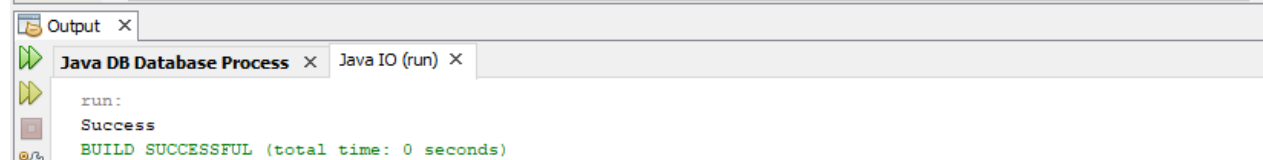
The bottom panel, titled 'Output', shows the execution results of the program:

```
run:
12352
25.5
A
Jim
BUILD SUCCESSFUL (total time: 0 seconds)
```

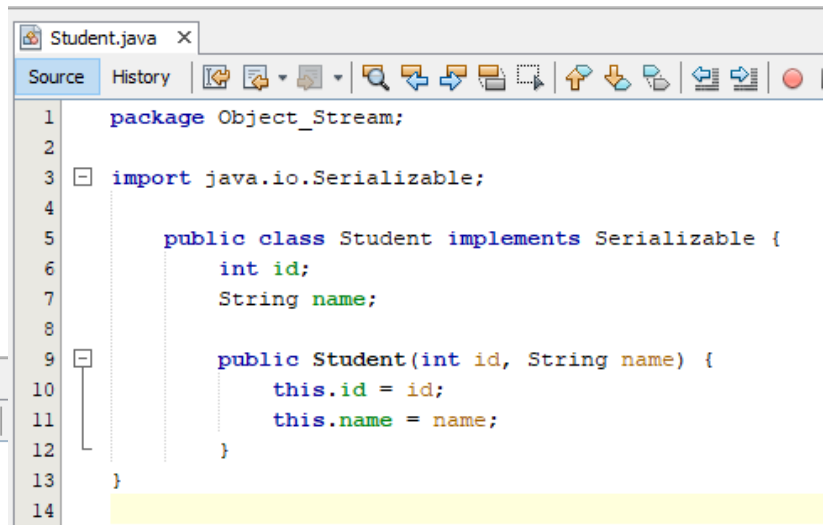
Object Stream



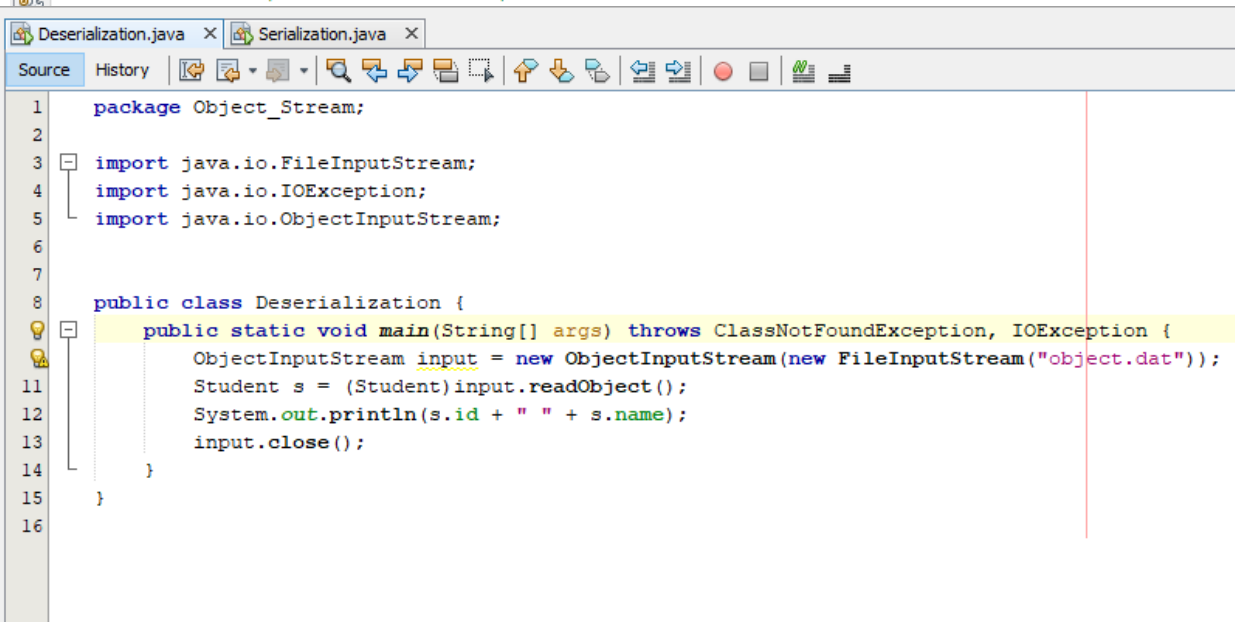
```
Deserialization.java x Serialization.java x
Source History
4 import java.io.IOException;
5 import java.io.ObjectOutputStream;
6
7
8 public class Serialization {
9     public static void main(String[] args) throws IOException {
10         Student s = new Student(1111, "Tae");
11         ObjectOutputStream output = new ObjectOutputStream(new FileOutputStream("object.dat"));
12         //
13         //
14         //
15         output.writeObject(s);
16         output.flush();
17         output.close();
18         System.out.println("Success");
19     }
20 }
21
```



```
Output x
Java DB Database Process x Java IO (run) x
run:
Success
BUILD SUCCESSFUL (total time: 0 seconds)
```



```
Student.java x
Source History
1 package Object_Stream;
2
3 import java.io.Serializable;
4
5 public class Student implements Serializable {
6     int id;
7     String name;
8
9     public Student(int id, String name) {
10         this.id = id;
11         this.name = name;
12     }
13 }
14
```

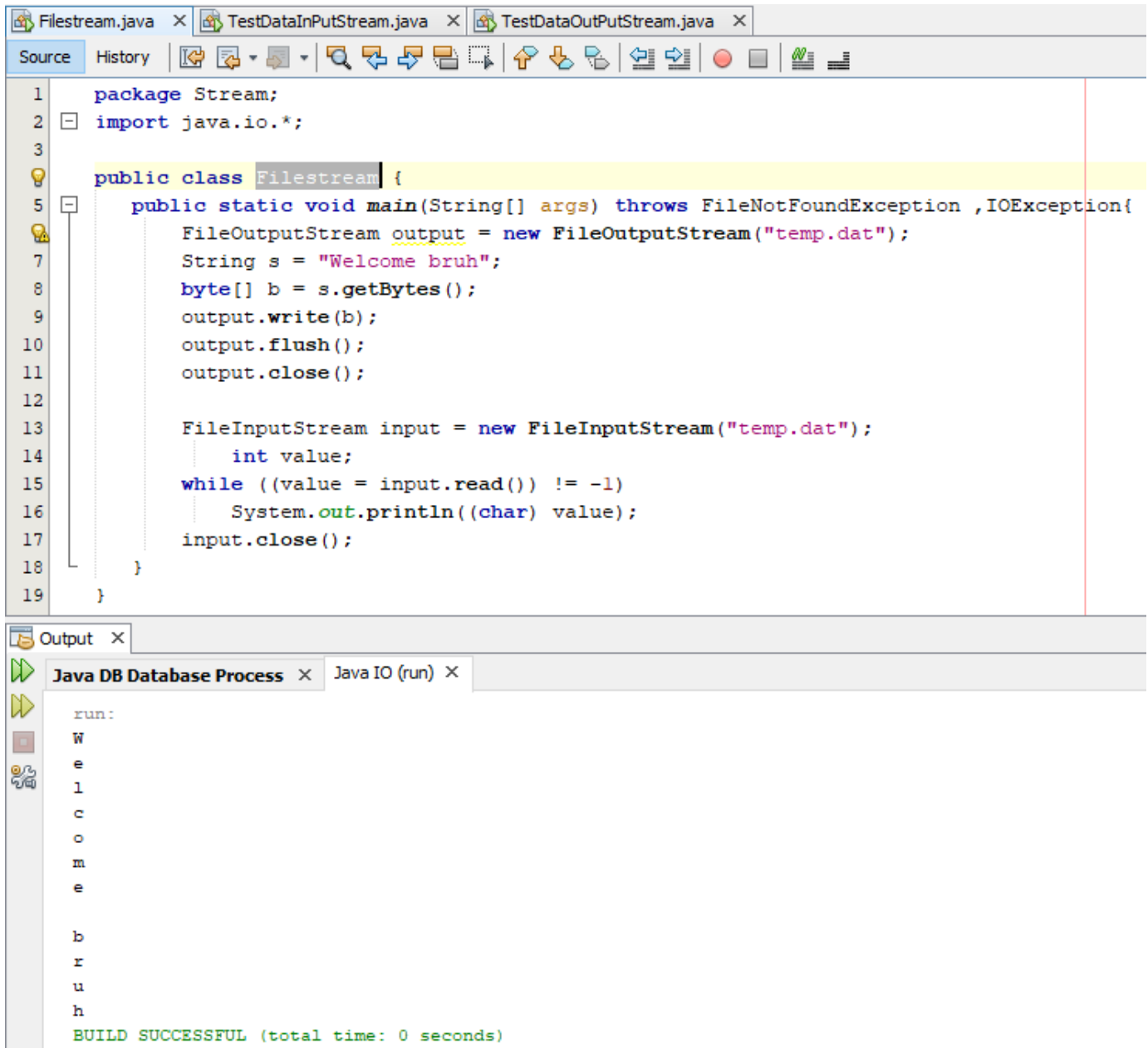


```
Deserialization.java x Serialization.java x
Source History
1 package Object_Stream;
2
3 import java.io.FileInputStream;
4 import java.io.IOException;
5 import java.io.ObjectInputStream;
6
7
8 public class Deserialization {
9     public static void main(String[] args) throws ClassNotFoundException, IOException {
10         ObjectInputStream input = new ObjectInputStream(new FileInputStream("object.dat"));
11         Student s = (Student)input.readObject();
12         System.out.println(s.id + " " + s.name);
13         input.close();
14     }
15 }
16
```



```
Output x
Java DB Database Process x Java IO (run) x
run:
1111 Tae
BUILD SUCCESSFUL (total time: 0 seconds)
```

Filestream



The screenshot shows an IDE with three tabs: Filestream.java, TestDataInPutStream.java, and TestDataOutPutStream.java. The Filestream.java tab is active, displaying the following code:

```
1 package Stream;
2 import java.io.*;
3
4 public class Filestream {
5     public static void main(String[] args) throws FileNotFoundException ,IOException{
6         FileOutputStream output = new FileOutputStream("temp.dat");
7         String s = "Welcome bruh";
8         byte[] b = s.getBytes();
9         output.write(b);
10        output.flush();
11        output.close();
12
13        FileInputStream input = new FileInputStream("temp.dat");
14        int value;
15        while ((value = input.read()) != -1)
16            System.out.println((char) value);
17        input.close();
18    }
19 }
```

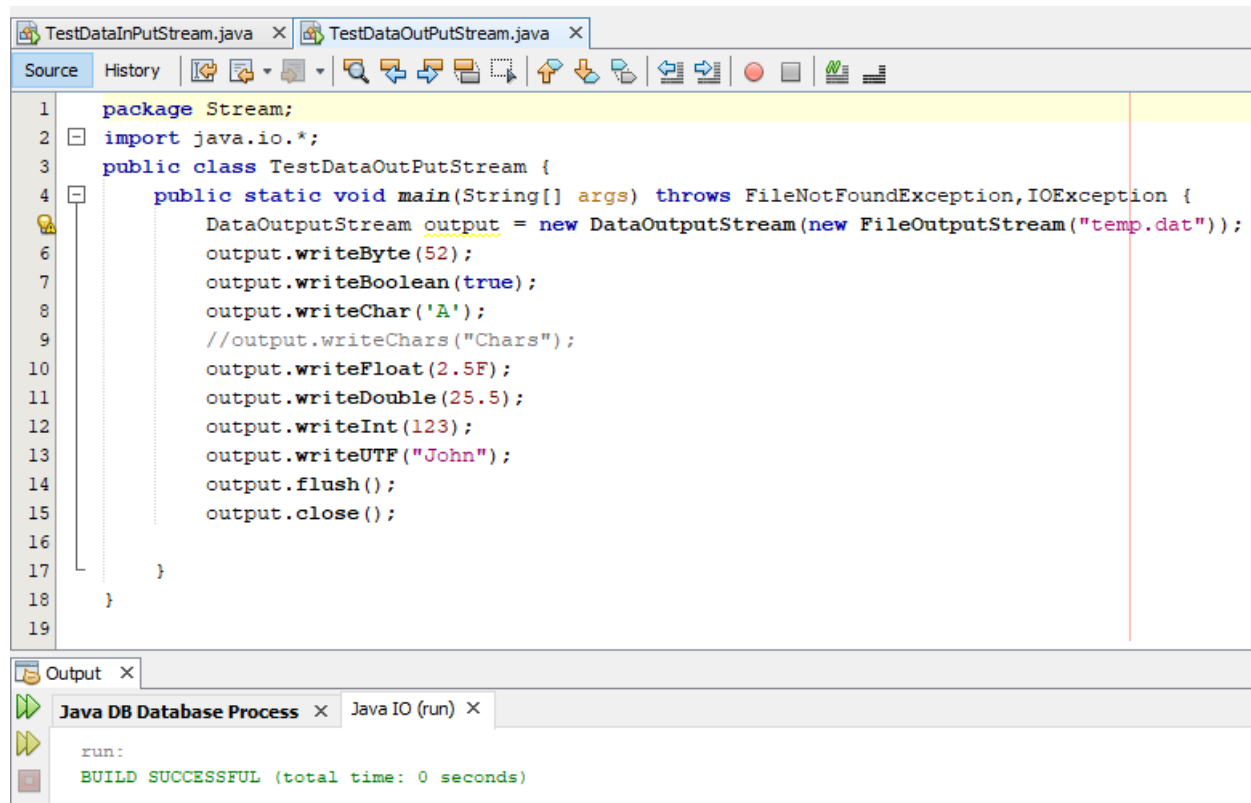
Below the code editor is the Output window, which shows the following output:

```
run:
W
e
l
c
o
m
e

b
r
u
h

BUILD SUCCESSFUL (total time: 0 seconds)
```

DataOutputStream



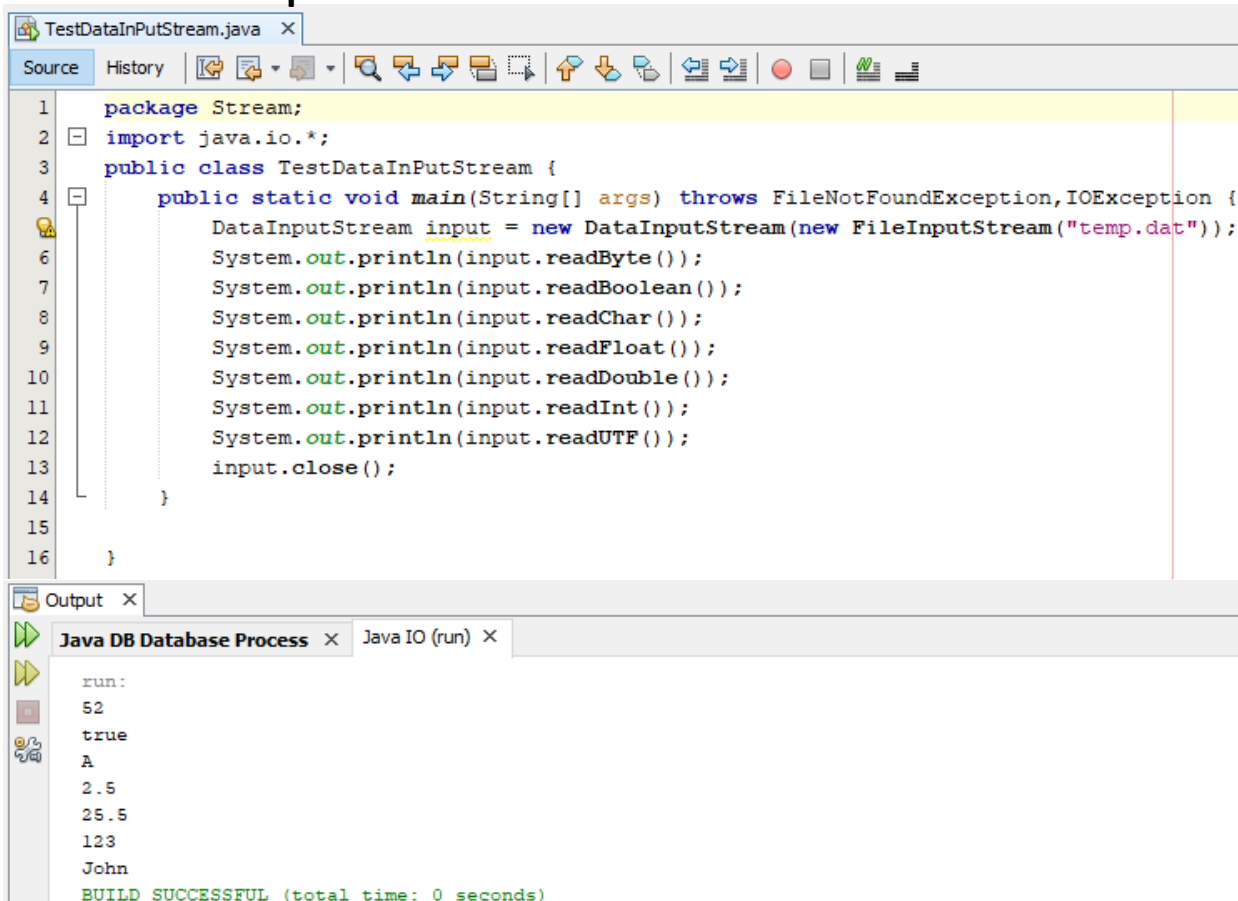
The screenshot shows an IDE with two tabs: `TestDataInPutStream.java` and `TestDataOutPutStream.java`. The `TestDataOutPutStream.java` tab is active, displaying the following code:

```
1 package Stream;
2 import java.io.*;
3 public class TestDataOutPutStream {
4     public static void main(String[] args) throws FileNotFoundException, IOException {
5         DataOutputStream output = new DataOutputStream(new FileOutputStream("temp.dat"));
6         output.writeByte(52);
7         output.writeBoolean(true);
8         output.writeChar('A');
9         //output.writeChars("Chars");
10        output.writeFloat(2.5F);
11        output.writeDouble(25.5);
12        output.writeInt(123);
13        output.writeUTF("John");
14        output.flush();
15        output.close();
16    }
17 }
18 }
19 }
```

Below the code editor, the `Output` window is visible, showing the results of the program execution:

```
run:
BUILD SUCCESSFUL (total time: 0 seconds)
```

DataInputStream



The screenshot shows an IDE with two tabs: `TestDataInPutStream.java` and `TestDataOutPutStream.java`. The `TestDataInPutStream.java` tab is active, displaying the following code:

```
1 package Stream;
2 import java.io.*;
3 public class TestDataInPutStream {
4     public static void main(String[] args) throws FileNotFoundException, IOException {
5         DataInputStream input = new DataInputStream(new FileInputStream("temp.dat"));
6         System.out.println(input.readByte());
7         System.out.println(input.readBoolean());
8         System.out.println(input.readChar());
9         System.out.println(input.readFloat());
10        System.out.println(input.readDouble());
11        System.out.println(input.readInt());
12        System.out.println(input.readUTF());
13        input.close();
14    }
15 }
16 }
```

Below the code editor, the `Output` window is visible, showing the results of the program execution:

```
run:
52
true
A
2.5
25.5
123
John
BUILD SUCCESSFUL (total time: 0 seconds)
```