

© 2003 Walter Savitch

These slides are for the exclusive use of students in
CSE 11 at UCSD, Winter quarter 2003.

They may not be copied or used for any other
purpose without the written permission of Walter
Savitch.

wsavitch@ucsd.edu

CSE 11

March 11, 2003

Final Exam

A00: Wednesday, March 19th

8 - 10:59am, Center 109 (note Room)

B00: Wednesday, March 19th

11:30 - 2:29pm, Solis 111

You must take it in the section you are enrolled in. We will check!

Must have a Picture ID--Same rules as midterm.

Closed book, One file card OK

Like Midterm but about twice as long.

Covers entire course.

Unwrapping SavitchIn

Code uses `System.in.read()` and
`System.in.readLine()`

These behave just like `read` and `readLine`
in `BufferedReader`.

```
public static char readChar( )
{
    int charAsInt = -1;
    try
    {
        charAsInt = System.in.read( );
    }
    catch(IOException e)
    {
        System.out.println(
            "Fatal error.");
        System.exit(0);
    }

    return (char)charAsInt;
}
```

**to get numbers use
StringTokenizer
Integer.parseInt
Double.parseDouble
etc.**

**Read code of SavitchIn.
May be on the final.**

Binary Files

Writing

```
try
{
    DataOutputStream outputStr=
        new DataOutputStream(
            new FileOutputStream(
                "numbers.dat"));
    int n;

    System.out.println("Enter");
    n = SavitchIn.readLineInt( );
    outputStr.writeInt(n);

    outputStr.close( );
}
catch(IOException e)
{
    System.out.println("Problem");
}
```

`writeInt`

`writeDouble`

`writeChar`

`etc.`

No `writeString`

Use `writeUTF`

UTF stands for Unicode Text Format.

Reading

```
try
{
    DataInputStream inputStr =
        new DataInputStream(
            new FileInputStream(
                "numbers.dat"));
    int n;
    n = inputStr.readInt( );
    System.out.println(n);
    inputStr.close( );
}
catch(FileNotFoundException e)
{
    System.out.println("Opps");
}
catch(IOException e)
{
    System.out.println("Problem");
}
```

Read and Write Must Match

writeInt	readInt
writeDouble	readDouble
writeUTF	readUTF
etc.	

Mixing data types in the same file is legal but requires some care.

Read to End of File

```
try
{
    while (true)
    {
        n = inputStream.readInt( );
        System.out.println(n);
    }
}
catch(EOFException e)
{ }
```

Serializeable (Not in text, not on Final)

**use ObjectOutputStream not
DataOutputStream**

**use ObjectInputStream not
DataInputStream**

**Write/Read one entire object with
writeObject(...)/readObject(...)**

**Class being written/read must implement
the Serializeable interface.**