Input and Output (IO)

CSCE 110

Drawn from James Tam's material
Output

• Displaying information on screen
• Done via the write and writeln statements
  – Write: displays the output ______________
    (the cursor remains on the line)
  – Writeln: displays the output ____________
    _____ (the ________ moves to __________)
Output

• Format (literal string of characters):

write ('a message');

OR

writeln ('a message');
Output Example

program outputExample1 (output);
begin
  write('line1');
  writeln('line2');
  write('line3');
end.

• You can find an copy of this program and the compiled version in the course directory: /pub/courses/csce110/

J. Michael Moore
Output the Contents Of Variables And Constants

• **Format:**

  write(<name of variable> or <constant>);

  or

  writeln (<name of variable> or <constant>);
Output Variables And Constants Example

program outputExample2 (output);
const
    ACONSTANT = 888;
begin
    var num : integer;
    num := 7;
    writeln(ACONSTANT);
    writeln(num);
end.

• You can find an copy of this program and the compiled version in the course directory:
  /pub/courses/csce110/
Mixed Output

• It's possible to display literal strings of characters and the contents of variables and constants with a single write or writeln statement.

• Format:

```
write('message', <name of variable>, 'message'...);
```

OR

```
writeln('message', <name of variable>, 'message'...);
```

J. Michael Moore
Mixed Output Example

program outputExample3 (output);
const
    ACONSTANT = 888;
begin
    var num : integer;
    num := 7;
    writeln('ACONSTANT: ', ACONSTANT);
    writeln('num=', num);
end.

• You can find an copy of this program and the compiled version in the course directory: /pub/courses/csce110/
Automatic Output Formatting

•Automatic formatting of output
  –Field width: The computer will insert enough spaces to ______________________________
  ______________________________.
  –Decimal places: For ____ numbers the data will be displayed in _____________________ form.

1 These values can be set to any non-negative integer (zero or greater).
Manual Output Formatting

• Format:
  write or writeln (<data>:<Field width for data\(^1\>):<Number decimal places for real data\(^1\)>);

• Examples:
  var num : real;
  num := 12.34;
  writeln(num);
  writeln(num:5:2);

---

1 These values can be set to any non-negative integer (zero or greater).

J. Michael Moore
Formatting Output

• If the field width doesn't match the actual size of the field
  – Field width too small – extra spaces will be added for ____________ variables but not for ________________.
  – Examples:

    var num : integer;
    num := 123456;
    writeln(num:3);
    writeln('123456':3);

J. Michael Moore
Formatting Output

• If the field width doesn't match the actual size of the field
  – Field width too large – the data will be _______ justified (extra spaces will be put ______________ the data).
  – Examples:

```pascal
var num : integer;
    num := 123;
    writeln(num:6);
    writeln('123':6);
```

J. Michael Moore
Formatting Output

• If the number of decimal places doesn't match the actual number of decimal places.
  – Set the ______________________ less than the _____________________________ – the number will be ______________________.
  – Example One:
    ```pascal
    var num : real;
    num := 123.4567;
    writeln (num:6:2);
    ```

J. Michael Moore
Formatting Output

• If the number of decimal places doesn't match the actual number of decimal places.
  – Set the __________________________ greater than the ___________________________ – the number ________________________________.
  – Example Two:
    ```pascal
    var num : real;
    num := 123.4567;
    writeln(num:12:6);
    ```
Input

• The computer program getting information from the user
• Done via the read and readln statements

• Format:

read (<name of variable to store the input>);

OR

readln (<name of variable to store the input>);
Input Example

program inputExampleOne (input, output);
begin
    var num : integer;
    write('Enter an integer: ');
    readln (num);
end.
Read & Readln

• Reads each ______ entered and matches it to the corresponding ___________.
  – e.g., read (num)
  – If num is an integer then the read statement will try to read an integer value from the user's keyboard input.
Read Vs. Readln

• Read
  – If the user ______________________ before hitting enter, the __________________________ __________________________.

• Readln
  – Any ______________________ before (and including) the enter key __________________________.
Read: Effect On The Keyboard Buffer

Pascal program

```
program getInput (input, output);
begin
    var num : integer;
    write('Enter an integer: ');
    read(num);
end.
```

*Keyboard*: user types in 27 and hits enter
Read: Effect On The Keyboard Buffer

### Pascal program

```pascal
program getInput (input, output);
begin
    var num : integer;
    write('Enter an integer: ');
    read(num);
end.
```

### Keyboard controller:

determines which keys were pressed and stores the values in the keyboard buffer

<table>
<thead>
<tr>
<th>Ram</th>
<th>num</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>27</td>
</tr>
</tbody>
</table>

Note: after the read statement has executed the pointer remains at the EOL marker.

1 When the user presses the enter key it is stored as the EOL (end-of-line) marker. The EOL marker signals to the Pascal program that the information has been typed in and it will be processed.

J. Michael Moore
Readln: Effect On The Keyboard Buffer

Pascal program

```pascal
program getInput (input, output);
begin
  var num : integer;
  write('Enter an integer: ');
  readln(num);
end.
```

Keyboard: user types in 27 and hits enter
Readln: Effect On
The Keyboard Buffer

Pascal program

```pascal
program getInput (input, output);
begin
  var num : integer;
  write('Enter an integer: ');
  readln(num);
end.
```

Keyboard controller: determines which keys were pressed and stores the values in the keyboard buffer.

<table>
<thead>
<tr>
<th>2</th>
<th>7</th>
<th>&lt;EOL&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>

Note: Unlike read, the readln will move the pointer past the EOL marker (input buffer is emptied and ready for new input).

1 When the user presses the enter key it is stored as the EOL (end-of-line) marker. The EOL marker signals to the Pascal program that the information has been typed in and it will be processed.

J. Michael Moore
Readln: Effect On The Keyboard Buffer

Pascal program

```pascal
program getInput (input, output);
begin
  var num : integer;
  write('Enter an integer: ');
  readln(num);
end.
```

Keyboard controller: determines which keys were pressed and stores the values in the keyboard buffer.

Note: Unlike read, the readln will move the pointer past the EOL marker (input buffer is emptied and ready for new input).

```
<table>
<thead>
<tr>
<th>RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>num</td>
</tr>
<tr>
<td>27</td>
</tr>
</tbody>
</table>
```

1 When the user presses the enter key it is stored as the EOL (end-of-line) marker. The EOL marker signals to the Pascal program that the information has been typed in and it will be processed.

J. Michael Moore
Read Vs. Readln

• If no input is read in by the program after a 'read' or 'readln' statement then both approaches appear __________ (the effect of the pointer staying or moving past the EOL marker has ____________________).

• Caution! If the 'read' or 'readln' statement is followed by another read or readln then the effect of the extra input remaining in the keyboard buffer can have ______________ ____________________!
Read Example

program read1 (input, output);
begin
  var num : integer;
  var ch  : char;
  write('Enter a number: ');
  read(num);
  write('Enter a character: ');
  read(ch);
  writeln('You entered num: ', num, ' ch: ', ch);
end.

• You can find an copy of this program and the compiled version in the course directory: /pub/courses/csce110/

J. Michael Moore
Readln Example

program read2 (input, output);
begin
    var num : integer;
    var ch : char;
    write('Enter a number: ');
    readln(num);
    writeln('You entered num: ', num);
    write('Enter a character: ');
    readln(ch);
    writeln('You entered ch: ', ch);
end.

• You can find a copy of this program and the compiled version in the course directory: /pub/courses/csce110/

J. Michael Moore
General Rule Of Thumb

• When getting input from the user unless there's a compelling reason you should use __________ rather than __________.

• (This is an important point: forget at your own peril!)
General Rule Of Thumb

• The prompt that requests user input should take the form of a _________ rather than a ________________:

```pascal
var num : integer;
write('Enter your age: ');  ______
readln(age);
```

```pascal
var num : integer;
writeln('Enter your age: ');  ______?
readln(age);
```
Another Use For Readln

• As an ______ prompt
• e.g.,

writeln('To continue press enter');
readln;
writeln('The rest of the program continues..');

When this statement is reached the program will ______
________________
________________

J. Michael Moore