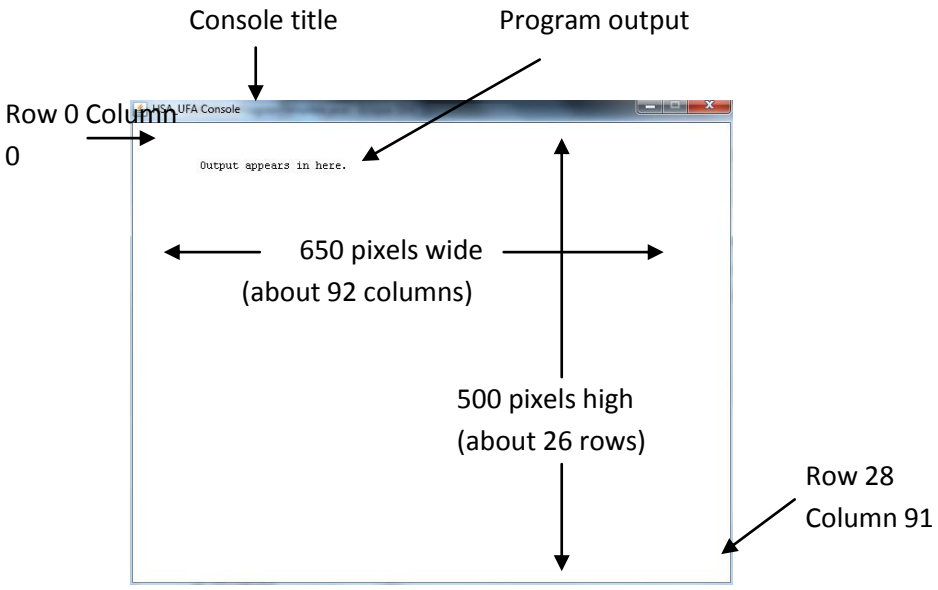
Console Text Output

Creating the Console Window

The GraphicsConsole window is the window that pops up when you run a program that uses the hsa2 console. The console window contains a text grid that by default holds 25 lines by 80 columns of text. The user can input data directly in the Console window and the output of the program appears there too.



Basic Text Output

For basic text output, the console is divided into **rows** and **columns**. The top left corner is row 0 and column 0. When you print something to the screen, where it appears is controlled by an invisible pointer called the **cursor**. The cursor starts in the top left corner (row 0, column 0) and moves to the right and down (increasing columns and rows) depending on how many characters you output.

The row number increases whenever you reach the end of a line, or when you output a **line feed** character. A line feed increases the row number by 1, and sets the column number to 0. When you reach the bottom of the screen, the output starts to scroll.

Text Output Methods

Important: Don't forget to put the name of the console and a dot before each of these when you use them in your programs (most of the time, it's "c.").

clear ()

Clears the Console window and sets the cursor to the upper-left corner (*row* = 0, *column* = 0).

> print() and println() are meant to imitate the built in Java commands System.out.print() and System.out.println() that print to the system console.

print (*String or other printable data*)

Prints whatever you put in the brackets, starting from the current cursor position and moving to the right (and down if you reach the end of the line). Usually, you put a **string** in the brackets, like "hello world" or "12345"... You can also print plain numbers without the quotation marks.

println ()

Outputs a line feed at the current cursor position, which moves the cursor down one line (*row* + 1) and all the way to the left (*column* = 0)

println (*String, or words in double quotation marks, or other printable stuff*)

Identical to print, except that a **line feed** is output after the argument. This means that the next time you use c.print or c.println the text will come out on the next line.

setColor (*Color*)

Sets the color for text output from *print* and *println* methods. See the Colors handout.

setBackgroundColor (*Color*)

Sets the background color for text output from print and println methods. See Colors handout.

setCursor (*row*, *column*)

Sets the cursor position to row *row* and column *col*.

close()

Closes the console window.

> If you put \n inside a string, it will go down to the next line. If you put \t, it will insert a tab. Try this:
> `c.print("Hello\nWorld.\tbye now");`

Quick Quiz

Draw the exact output for each of these two sequences of statements.

```
c.print("Hello ");
c.print("world. ");
c.println("Goodbye. ");
```

```
c.println("Hello ");
c.println("world. ");
c.println("Goodbye. ");
```

Warnings for common mistakes

1. After the line that says what your package is in your program, you'll have to include the following:
   import hsa2.GraphicsConsole;
   See ProgramTemplate.java in the pdf folder.

2. Make sure that you use straight quotes. " " not “ ”. This is often a problem when you cut and paste from internet documents.

3. The spelling of "colour" in programming is the American spelling.
   So you can create a variable: String textColour = "red"; since variables can be any spelling,
   but if you use any built in objects or functions, you have to write
   ```
   Color myBackground = Color.RED;
   c.setColor( new Color(100,200,255) );
   ```

4. println is lowercase for PRINTLN It's an L, not a 1 nor an i (as in print line)

5. **setBackgroundColor (*Color*)**
   The background colour is only visible after you clear the screen again, because clearing the screen actually redraws everything on it, including the background.
   So, always type: **c.setBackgroundColor (*Color*);** then **c.clear();**