Programming lesson #1a "C in 4 pages"

1) This is a 'C' program:

```
#include <stdio.h>          <- header file

#define PI 3.14159          <- #define

int main(void)
{
    int x = 0;

    printf("%d\n", x);
    printf("%f\n", PI);
}
```

variable

function

print function

How to format

what to print

2) a 'C' program is just text in a particular style, set of rules

3) need to translate it to "machine code" using a compiler (gcc) before it can be run ("executed") on machine:

```
gcc test.c -o test
```

4) To run this program:

```
./test
```

4a) modify test.c to print out your name, age.

Programming Lesson #16 'C' code continued

5) Here's another 'C' program:

```
#include <stdio.h>

#define PI 3.14159

int main(void)
{
    int x = 15;

    if( x > 10 )
    {
        printf("x is big\n");
    }
    else
    {
        printf("x is puny\n");
    }
}
```

semi-colon at end of statement

NO semi-colon at end of flow control command

5a) add "if()" test for case where x > 100 and print something different out.

Programming Lesson #1c 'C' code continued

6) Another 'C' program:

```
#include <stdio.h>

#define PI 3.14159

int main (void)
{
    int i;

    for(i=0; i<1000; i++)
    {
        printf("%d\n", i);
    }
}
```

"for" loop
ⓐ initializer
ⓑ test
ⓒ body
ⓓ increment

this executes ⓐ once and then ⓑ, ⓒ, ⓓ over and over, in that order.

6a) change "for" loop to count to 1 million.

6b) change printf command to also print i squared.

Programming Lesson #1d 'C' code continued

7) last 'C' program (for now :-)

```
#include <stdio.h>

#define PI  3.14159

float negate ( float value)
{
    return ( -1.0 * value);
}

int main ( void)
{
    float x = PI;
    float y;

    y = negate(x);

    printf ("%f\n", y);
}
```

input

output → negate

function

execution starts at main()

pass input

catches output

7a) add new function: float square (float value); that returns value squared.

7b) add printf to display "y" and "y" squared.