Lab 1

Crafting A Compiler

3.1

```
main(){
        const float payment = 384.00;
        float bal;
        int month=0;
        bal = 15000;
        while (bal>0){
                printf(“Month: %2d Balance:%10.2f\n”, month bal);
                bal = bal - payment + 0.015*bal;
                month= month+1;
        }
}
```

TOKEN STREAM:

main
right paren
left paren
left bracket
const
float
var (value of payment)
assignment
digit (value of 384.00)
;
float
var (value of bal)
;
int
var (value of month)
assignment
digit (value of 0)
;
var (value of bal)
assignment
digit (value of 15000)
;
while
left paren
var (value of bal)

Lab 1

>
digit (value of 0)
right paren
left bracket
printf
left paren
String (value of “Month: %2d Balance:%10.2f\n”)
var (value of month)
,
var (value of bal)
right paren
;
var (value of bal)
assign
var (value of bal)
-
var (value of payment)
+
digit (value of 0.015)
*
var (value of bal)
;
var (value of month)
assignment
var (value of month)
+
digit (value of 1)
;
right bracket
right bracket

Dragons

1.1.4
The advantage to using C as a target language for a compiler is that C is platform independent, runs fast, and is very malleable to do what you want it to.

Lab 1

1.6.1

```
int w, x, y, z;
int i = 4; int j = 5;
{   int j = 7;
    i = 6;
    w = i + j;
}
x = i + j;
{   int i = 8;
    y = i + j;
}
z = i + j;
```

(a) Code for Exercise 1.6.1

w=13
x= 11
y=13
z=11
i=6