CpS 230 Homework 3: Stack Frames

Specially Prepared for Class Example (class)

On the last page of this assignment, you will find a code listing for a (randomly generated) C program. Use it to answer all questions on the worksheet. Assume that the C compiler uses exactly the same (simple) rules for stack frame construction that we have studied in class.

1 Stack Frame Chart

Fill out the following table so that it reflects the values on the stack at the moment line 6 is about to be executed.

1. The "value" column should contain
   (a) a numeric value (decimal) if the value can be known, or
   (b) the string "???" if the value cannot be derived from the given information
2. The "description" column should contain
   (a) the name of the parameter/local variable stored in that slot, or
   (b) a description of its special role (e.g., 'saved EBP', 'return address')

Address	Value	Description
11100	???	saved EBP
11096		
11092		
11088		
11084		
11080		
11076		
11072		
11068		

2 Instruction Operands

Provide the missing operands for the following assembly instructions. Remember: in real life you will not know in advance the actual addresses at which parameters and local variables live, so you must use frame-pointer-relative addressing (i.e., `EBP + nn` or `EBP - nn`).

```
; Implementing line 16
push    dword [          ] ; Pass jackal
push    dword [          ] ; Pass bat
call    _lime
add     esp, 8             ; Remove parameters from stack
mov     [          ], eax  ; Move return value into cat

; Implementing line 6
mov     eax, [           ]  ; Get fire_engine
add     eax, [           ]  ; Combine with drill
add     esp, 4              ; Release local variable storage
pop     ebp                 ; Restore previous frame pointer
ret                         ; Return to caller
```

3 Source Code

```
1  int lime(int fire_engine, int pogo_stick) {
2      int drill = 3400;
3
4      // ...
5
6      return fire_engine + drill;
7  }
8
9  int main() {
10     int bat = 7400;
11     int jackal = 4200;
12     int cat = 8200;
13
14     // ...
15
16     cat = lime(bat, jackal);
17
18     // ...
19
20     return 0;
21 }
```