CS 301 Midterm Exam answer key

0.) NAME: ________answer key___________________
2015-10-21. Closed book, closed laptop, closed notes.

1.) Fill in the blanks to make true statements, for a 64-bit Linux machine (like NetRun's default).

- A function's return value is stored ___in register rax____________________________.
- A function's first parameter is stored _____ in register rdi _______________.
- A class with one long uses ____ 8 ____ bytes of memory.
- You would write a C++ prototype for an assembly function named "bar" with no parameters or return value as:
  _____extern "C" void bar(void); _____

(Score: 10 points.)

2.) Convert each piece of C++ code to assembly, and vice versa:

C++	***Assembly***

long grabit() {
return 5;
}
	`grabit:`
	`mov rax,5`
	`ret`

// The functions phase1 and phase2
// both take and return one long.
long pipeline(long x) {
return phase2(phase1(x)+3);
}
	`pipeline:`
	`; x stored in rdi`
	`call phase1`
	`mov rdi,rax`
	`add rdi,3`
	`call phase2`
	`; return value in rax`
	`ret`

(Score: 30 points. Each piece of code is separate.)

3.) There are several memory allocation styles. Fill in the remaining table entries.

What?	*Why?*	*How? (allocate one long)*	*Static or Dynamic?*
section .data	Named static variables	`dq 3`	static
the stack	Fast function-local allocation	sub rsp,8	dynamic
malloc	Long-lived allocations	mov rdi, 8 call malloc	dynamic

(Score: 10 points. "Static" allocation happens once per program; "Dynamic" can happen again and again.)

4.) Fix the errors in this assembly translation of this C++: **`for (int i=0;i<n;i++) arr[i]=3;`**

```
    ; input: rsi == arr, rdi == n
    mov rcx,0 ; i
    jmp check ; need to verify i<n before starting loop
start:
    mov DWORD[rsi + rcx *4], 3  ; need to scale for array access
                                    (DWORD is 4 bytes each)
    add rcx,1
check:
    cmp rcx,rdi
    jle start   ; just jl, jle is like i<=n
```

(Score: 15 points.)

5.) You see the compiler uses "push rbx" at the start of your function, and "pop rbx" at the end. Why?

rbx is a preserved register, and the compiler wants to store something in it.

(Score: 10 points.)

6.) If each piece of code returns a value, write the value. If it'll crash or hang, write that, and why.

`push rax`
`push 3`
`pop rax`
`ret`
	`push 2`
	`push 7`
	`pop rax`
	`pop rcx`
	`ret`
		`mov rax, 9`
		`push rax`
		`sub rsp, 8`
		`ret`
			`push 6`
			`mov rax,QWORD[rsp]`
			`pop rcx`
			`ret`
				`mov rax,0x3F`
				`and rax,0x55`
				`ret`

Crash (not enough pops)	7	Crash (moved stack pointer wrong way; add rsp,8 would work)	6	0x15

(Score: 15 points. Several of these are ... subtle. Be careful.)

7.) You had asked your newest employee to measure the speed of an empty function on Intel's new server chip, and his entire email in response was "503,148.7 per second". What's wrong with his response?

1.) No units—should be "xxx function calls per second, as measured from C++."
2.) No error bars—should be "xxx +- yy", or even give a standard deviation or histogram of measurement variance.

(Score: 10 points.)