Tutorial 11
CSE 112 Computer Organisation

Q1. What is caller-callee convention, and why is it needed?
Q2. Assuming that each function is equally likely to use every register, what should be the ideal split between the number of callee-saved and caller-saved registers? Q3. What is a stack?

ISA Description and Caller-Callee conventions for Q4:

Name	**Semantics**	**Syntax**
Add	Performs rd = rs1 + rs2	add rd rs1 rs2
Sub	Performs rd = rs1 - rs2	sub rd rs1 rs2
Add Immediate	Performs rd = rs + Imm	addi rd, rs, #Imm
Branch if not equal	If content of rs1 is not equal to content of rs2 then branch to label	bne rs1 rs2 **Label**

Jump and link	Jumps to label after saving the return address to rd.
	rd=PC+4
	PC=PC+sext(imm[20:1], 1 ‘b0}
		jal rd, **Label**

Jump and Link Register	Jump to the address formed by adding the source register and immediate after saving the return address to rd.
	rd=PC+4
	PC = rs1 + sext(imm[11:0])
		Jalr rd,rs1,imm[11:0]

7.2.8 Register Encoding

Address	Register	ABI Name	Description	Saver
5'b0000_0	x0	zero	Hard-wired zero	– – – –
5'b0000_1	x1	ra	Return address	Caller
5'b0001_0	x2	sp	Stack Pointer	Callee
5'b0001_1	x3	gp	Global Pointer	– – – –
5'b0010_0	x4	tp	Thread Pointer	– – – –
5'b0010_1	x5	t0	Temporary/alternate link register	Caller
5'b00_{110,111}	x6-7	t1-2	Temporaries	Caller
5'b0100_0	x8	s0/fp	Saved register/frame pointer	Callee
5'b0100_1	x9	s1	Saved Register	Callee
5'b0101_{0,1}	x10-11	a0-1	Function arguments/ return values	Caller
(5'b011_{00-11}),(5'b1000_{0,1})	x12-17	a2-7	Function arguments	Caller
5'b1_{0010-1011}	x18-27	s2-11	Saved registers	Caller
5'b111_{00-11}	x28-31	t3-6	Temporaries	Caller

- **There are 32 general purpose registers x0-x31.**
- **ra is the return address register.**
- **a0,a1 stores the return value.**
- **The first two arguments to the callee are stored in registers a2 and a3. ● All the registers from s2-s11 are caller-saved. On the other hand, registers s0-s1 are callee saved.**
- **Whenever the branch and link instruction is used, the return address is stored in ra, and the program counter jumps to the given label.**

Q4.Given the C code, convert the following into assembly language

```
void bar()
{
        c = baz(1, 2);
        d = c+3
}
int baz(int a, int b)
{
        return a + b;
}
```