1.

A.) Name all eight 32-bit general-purpose registers.

EAX	ECX	EBP	ESI
EBX	EDX	ESP	EDI

B.) What is the general function of each of the registers?

The general-purpose registers are primarily used for arithmetic and data movement, but some registers have specialized uses.

EAX - Automatically used by multiplication and division instructions. It is often called the extended accumulator register. (Accumulator)

ECX - The CPU automatically uses it as a loop counter. (Loop counter)

ESP - Addresses data on the stack (a system memory structure). It is rarely used for ordinary arithmetic or data transfer. It is often called the extended stack pointer register. (Stack pointer)

ESI – used by high-speed memory transfer instructions. They are sometimes called the extended source index and extended destination index registers. (Index register)

EDI - used by high-speed memory transfer instructions. They are sometimes called the extended source index and extended destination index registers. (Index register)

EBP – used by high-level languages to reference function parameters and local variables on the stack. It should not be used for ordinary arithmetic or data transfer except at an advanced level of programming. It is often called the extended frame pointer register. (Extended frame pointer (Stack))

C.) Which of these registers can be addressed in parts?

EAX	EBX	ECX	EDX

2. What do the Sign Flag, Zero Flag, Auxiliary Carry Flag, and Parity Flag indicate when set?

Sign Flag (SF)	It is set when the result of an unsigned arithmetic operation is too large to fit into the destination. (Result is negative.)
Zero Flag (ZF)	It is set when the result of an arithmetic or logical operation generates a result of zero. (Result is zero.)
Auxiliary Carry Flag (AC)	It is set when an arithmetic operation causes a carry from bit 3 to bit 4 in an 8-bit operand. (Carry from bit 3 to bit 4.)
Parity Flag (PF)	It is set if the least-significant byte in the result contains an even number of 1 bits. Otherwise, PF is clear. In general, it is used for error checking when there is a possibility that data might be altered or corrupted. (Sum of 1 bits is an even number.)

3.

A.) What do the Overflow Flag and Carry Flag indicate when set?

Overflow Flag (OF)	It is set when the result of a signed arithmetic operation is too large to fit into the destination. (Signed arithmetic out of range.)
Carry Flag (CF)	It is set when the result of an unsigned arithmetic operation is too large to fit into the destination. (Unsigned arithmetic out of range.)

B.) How do they differ?

The main difference is that one is used for only signed arithmetic, which is the Overflow flag, and the other is used for unsigned arithmetic.

4. Detail the process by which instructions and data are read from memory.

Reading from memory requires multiple machine cycles. The steps are:

1. Address placed on address bus.
2. Read Line (RD) set low.
3. CPU waits one cycle for memory to respond.
4. Read Line (RD) goes to 1, indicating that the data is on the data bus.

5. How is multitasking achieved?

Multitasking is achieved by using a component of the operating system named the scheduler which allocates a slice of CPU time (time slice) to each task. During a single time slice, the CPU executes a block of instructions, stopping when the time slice has ended. The processor saves the state of each task before switching to a new one. A task's state consists of the content of the processor registers, program counter, and status flags, along with references to the task's memory segments. The OS will usually assign varying priorities to tasks, giving them relatively larger or smaller time slices. By rapidly switching task, the processor creates the illusion they are running simultaneously.

6. Let us say your computer has only 512MB available for your process. But your program needs 1GB of memory. How does the computer make it possible to execute your program, as well as the other processes?
Your computer uses Paging, where it divides segments into blocks called pages. When a task is running, parts of it can be stored on disk if they are not currently in use. Parts of the task are paged (swapped) to disk. Other actively executing pages remain in memory. When the processor beings to execute code that has been paged out of memory it issues a page fault, causing the page or pages containing the required code or data to be loaded back into memory.

7.

A.) What do you understand by Real-address mode, Protected mode, Multi-segment model, and Paging?

1. Real-address mode:
   Real-address mode only provides 1MByte of memory that can be addressed. The processor can run only one program at a time, but it can momentarily interrupt that program to process request from peripherals.
2. Protected mode:
   Protected mode only provides 4Gbyte of memory to each process that is running. The processor can run multiple programs at the same time. Each program can be assigned its own reserved memory area, and programs are prevented from accidentally accessing each other's code and data.
3. Multi-segment model:
   In the multi-segment model, each program is given its own table of segment descriptors, called a local descriptor table (LDT). Each descriptor points to a segment, which can be distinct from all segments used by other processes. Each segment has its own address space.
4. Paging:
   Paging is a feature that permits segments to be divided into 4,096 byte blocks of memory called pages. Paging permits the total memory used by all programs running at the same time to be much larger than the computer's physical memory. The complete collection of pages mapped by the operating system is called virtual memory.

B.) How does Real-Address mode differ from Protected mode?
Real-Address mode differs from Protected mode by one having less memory that is allotted to the program. Real-Address mode provides 1MByte and Protected Mode provides 4GByte. Also Real-Address mode allows direct access to system hardware whereas Protected mode does not. Also Real-Address mode can access any memory location whereas protected mode each program is assigned their memory location and can be assigned a reserved memory area, but programs are prevented from accessing each other's code and data.

C.) What are the defining attributes of a Multi-segment model? Discuss in detail.

In the multi-segment model, each program is given its own table of segment descriptors, called a local descriptor table (LDT). Each descriptor points to a segment, which can be distinct from all segments used by other processes. Each segment has its own address space. In each entry in the LDT points to a different segment in memory. Each segment descriptor specifies the exact size of its segment.

8. In a 32-bit computer what are the maximum memory that can be addressed in (a) real-addressed mode (b) protected mode?

A.) Real-addressed mode:

Maximum memory that can be addressed is 1 MByte.

B.) Protected Mode:

Maximum memory that can be addressed is 4 GByte.

9. What is the linear address corresponding to the following segment-offset: 08C4:1141?

The linear address corresponding to the above segment-offset is:

$$segment * 10h = adjusted\ segment\ value$$
$$Adjusted\ segment\ value + offset\ value = linear\ address$$

		0	8	C	4	Segment
			*	1	0	10_{16} (16_{10})
		0	0	0	0	
+	0	8	C	4	0	Adjusted segment value
		8	C	4	0	
	+	1	1	4	1	Offset value
		9	**D**	**8**	**1**	**Linear Address**

10. Let us say your computer is running at 4.8 GHz. You come to know that the Add instruction takes 5 clock periods in your computer. Express the time taken by the Add instruction in nanoseconds.

$$T = \frac{1}{f}$$

$$\frac{1}{4.8 * 10^9} = 2.08333 * 10^{-10} = .208333\ nano\ seconds.$$

$$5(.208333 * 10^{-9}) = 1.041665 * 10^{-9} = 1.041665\ nano\ seconds.$$