This activity needs all the students to create 10 minimum questions from the following topics.

1. **text section**
2. **data section**
3. **bss**
4. **stack**
5. **call stack**

1) Text Section

1. Give another name for Text Segment.

- Code Segment

2. Why we use Text Segment?

- To store assembled machine language instructions

3. To where the EIP register is pointing within the text segment?

- To the first instruction in the segment

4. What processes are done within an execution loop?
   1) Reads the instruction that EIP is pointing to.
   2) Adds the byte length of the instruction to EIP.
   3) Executes the instruction that was read in step 1.
   4) Goes back to step 1.

5. What will happen if you don’t update the instruction by adding byte length?

- First instruction will always be executed.

6. Why does it allow the EIP to go to another different memory area & again come back to the original place?

- Because it does not expect linear execution.

7. Why “write” permission is disabled in text segment?

- Because it’s not used to store variables but to store only code.

8. What are the advantages in disabling “write” permission?

- It prevents people from modifying the program code.
- It allows multiple executions of the program at the same time without any problems.

2) Data Section

1. Give another name for Data Segment.

- Initialized Data segment

2. The data segment is filled with the initialized _______, _______ and _______variables.

- global, static, constant, external

3. Data segments can further be classified as _______ and _______.

- Read-only area, read-write area

4. The variable initialized as “const” will come under ________ area. Remaining all others will come under ________ area.

- Read-only, read-write

3) BSS

1. Give another name for BSS.

- Uninitialized Data segment

2. The bss segment is filled with their _______________.

- Uninitialized counterparts

3. What does BSS refers to?

- Block Started by Symbol

4) Stack

1. Where does stack grows towards?

- Lower Address Space

2. What arguments are stored on top of the stack memory area?

- Command-line arguments & environmental variables

3. Why we use Stack?

- While your program is running, the machine language instructions load into the stack.

4. All ________________ are stored in stack segment.

- Local variables

5. Stack segment is used for passing __________ to the __________ along with the return address of the instruction.

- Arguments, functions

6. Give some uses of stack?

- Remember all of the passed variables.
- The location the EIP should return to after the function is finished.

7. Stack refer ______________ order.

- Last-in First-out (LIFO) / First-in Last-out (FILO)

8. When an item is placed into the stack it's called as ________ and when an item is removed from the stack it's called as _________.

- Pushing, popping

9. What is the purpose of having ESP register?

- To keep track of address of the end of the stack.

10. Because of its _________ behavior, we can consider that stack doesn't have a fixed size.

- Dynamic

11. What stack frames can be seen in here?

- Main Frame
- Secret Frame

5) Call Stack

1. Explain the process that happens in here.

- When you call a function, the system set aside space in memory for that function to do its necessary work.

2. Here we frequently call _____________ stack frames or __________.

- Chunks of memory, function frames

3. “More than one function’s stack frame may exist in memory at a given time.” True or false?

- True

4. For a given time, how many functions (open frames) can be active?

- Only one of those frames

5. How are these frames arranged?

- In a stack

6. The frame for the most-recently called function is always ___________.

- On top of the stack

7. In which locations the functions / frames are located inside the stack?

- main () → At the bottom
- move () → Above the main ()
- direction () → On the top - (This is the active frame now)