# Computer Science - ICS Part 2 Computer Science Chapter 8 Short Questions Preparation

Q1. What is Assembly Language?

**Ans 1:** Assembly level language is a set of codes that can run directly on the computer's processor. This type of language is most appropriate in writing operating systems and maintaining desktop applications. With the assembly level language it is easier for a programmer to define commands. It is easier to understand and use as compared to machine language.

---

Q2. Differentiate between Logical Errors and Syntax Errors.

**Ans 1:** Logical Errors : Logical errors when a program follows a faculty algorithm. The compiler cannot detect logical errors; therefor no error message is reported from the compiler . Moreover, these errors don't cause the program to be crashed ,that's why these are very difficult to detect. One can recognize logical errors by just looking at the wrong output of the program. Logical errors can only be detected by through testing of the program.

**Ans 2:** Syntax Errors: A Syntax error occurs when the program violates one or more grammar rule of C language.The compiler detects these errors as it attempts to translate the program. If a C statement has syntax error, it cannot be translated and the program could not be executed.

---

Q3. Differentiate Between Linking and Loader.

**Ans 1:** **Linking :**The Linker is a program the combine the object program with additional object files that may be needed for the program to execute and save the final machine language program as an executable file on disk. The linker combine different library files to the object file and produces an executable file with exe extension

**Ans 2:** **Loding :** A loader takes an executable file and copies its section into memory. Then it produces a process control block to control program execution. Finally, it starts executing the code , usually by jumping to its main address .

---

Q4. Define Computer Program?

**Ans 1:** A computer is a device that follows the instructions given to it . A well define set of instruction given to the computer is called a computer program.

---

Q5. How program logic is implemented in un-structured programming languages?

**Ans 1:** In un-structured programming languages, the entire logic of the program is implemented in a single module (function), which cause the program error prone, difficult ti understand, modify and debug.

---

Q6. Define the term Debug.

**Ans 1:** Debugging is the routine process of locating and removing program bugs, errors or abnormalities, which is methodically handled by software programmers via debugging tools. Debugging checks, detects and corrects errors or bugs to allow proper

program operation according to set specifications.

---

Q7. What is the difference between High level language and low Level language?

**Ans 1: High Level Language :** Program languages whose instructions resemble the English Language are called high level languages. Every high level language define a set of rule of writing. Programs called syntax of the language. Every instruction in the high level language must confirm to its syntax .

**Ans 2: Low Level Language :**A low level language is a programming language that deals with a computers hardware components and constraints. It has no (or only a minute level of) abstraction in reference to a computer and works to manage a computers operational semantics. A low-level language may also be referred to as a computer native language.

---

Q8. Define Runtime Errors.

**Ans 1:** A Runtime errors occur when the program directs the computer to perform an illegal operation, such as dividing a number by zero. Runtime errors are detected and displayed by the computer during the execution of a program. When a runtime error occurs the computer stops executing the program and displays a diagnostic message.

---

Q9. What is define directives?

**Ans 1:** The define create a macro, which is the association of an identifier or parameterized identifier with a token string, after the macro is defined, the compiler can substitute the token string for each occurrence of the identifier in the source file.
Syntax : #define identifier token -stringopt
#define identifier ( identifieropt,..., identifieropt) token-stringopt

---

Q10. Define Constant Macro.

**Ans 1: Constant Macro:** Constant Macro is a name that is replaced by a particular constant value. Before the program is sent to the compiler.
Syntax : #define Macro_Name expression
Example: #define PI 3.142857 # define SEC_PER_HR 3600

---