1. “software engineering is part of system Engineering process”. Do you agree with the above statement? Justify your above.

Answer

**Yes,**

Software engineering is concerned with all aspects of software production from the early stages of system specification through to maintaining the system after it has gone into use.

When system engineering is concerned with all aspects of computer-based systems development including hard ware, software and process engineering. Combined with the evolution of complex systems where software plats a major role.

There we can easily conclude that the software engineering is part of this more general system engineering processes.

1. **what is “software crisis”?**

**“software crisis** “it’s the term used in the field of computer science for the difficulty of writing useful and efficient computer programs in the required time. This was due to using sim work force, same work force, same method, and same tools after fast increasing in software demand and software challenges, then they arise some problems like software budget, software quality etc. this condition is called software crisis.

1. **what are the professional responsibilities of a software engineer?**

below are some major responsibilities of a software,

1). **Competence**

You should normally respect the confidentiality of your employers or clients irrespective of whether or not a formal confidential agreement has been signed.

2). **competence**

You should nit mis respect your level of competence. You should not knowingly accept work that is outside your competence.

3). **Computer misuse**

you should not use technical skills to misuse other people’s computers. Computer misuse ranges from relatively trivial (game playing on an employer’s machine) to extremely serious (dissemination of viruses or other malware).

1. **“competent-based software engineering allows faster delivery”. State whether this statement is true or false, justify your answer**

True

Competent based software engineering allows to develop faster and more reliable systems by reusing previously tested components.

With the use if “reuse of oriented components” the major advantage is reducing the amount of software to be developed. Usually, smaller codebases are easier to maintain since the services are separate. Therefore, reduced cost and risk, and usually leads to faster delivery.

In addition, it’s easy to test specific “input for output”. So, no longer have to write endless lines of codes to build an application with the help of component-based engineering.