TOPIC 1. Fundamentals of Software Engineering

Exercise 1. *Memorize the following words and word combinations.*

contentious	дискусійний, спірний
practitioner	виконавець; діяч; той, хто робить що-небудь постійно
provoke	викликати, спонукати
perceive	відчувати, почувати
muddied	заплутаний, незрозумілий
cue	знак, сигнал
dispute	диспут, дебати
complementary	додатковий, доповняльний
solely	тільки, виключно
come up with	порівнятись з, пропонувати (план, проект)
acquisition	придбання, надбання
impediment	перешкода, недолік
assessment	оцінка, оцінювання
reengineering	відновлення, реконструювання
pre-requisite	необхідна умова
master	опановувати
undergraduate	студент, новачок, початківець
internship	інтернатура
encounter	стикатися, зіткнутися
downright	явний, прямий
likewise	подібно, більше того

Exercise 2. *Read and translate the text.*

Text 1. Software Engineering as a contentious issue

Initially, when the first modern digital computers appeared in the early 1940s, the instructions to make them operate were wired into the machine. Practitioners quickly realized that this design was not flexible and came up with the "stored program architecture". Thus the first division between "hardware" and "software" began with an abstraction being used to deal with the complexity of computing.

The term "software engineering" first appeared in the 1968 NATO Software Engineering Conference and was meant to provoke thought regarding the perceived "software crisis" at the time. The Computer Society's Software Engineering Body of Knowledge defines "software engineering" as the application of a systematic, disciplined approach to the development, operation and maintenance of software, and the study of these approaches. It is the application of Engineering to software because it integrates significant mathematics, computer science and practices whose origins are in Engineering.

The relationship between computer science and software engineering is a contentious issue, which is further muddied by debates within software engineering disputes over what the term "software engineering" means, and how computer science is defined. David Parnas, taking a cue from the relationship between other engineering and science disciplines, has claimed that the principal focus of computer science is studying the properties of computation in general, while the principal focus of software engineering is the design of specific computations to achieve practical goals, making the two separate but complementary disciplines: "Rather than treat software engineering as a subfield of computer science, I treat it's as an element of the set: Civil Engineering, Mechanical Engineering, Chemical Engineering, Electrical Engineering ...". Thus, different efforts to solve this complicated problem have not giving a necessary result.

Time has seen significant improvements in the using and effectiveness of computer technology. Modern society has seen a significant shift from computers being used solely by experts or professionals to a more widespread user base. You are absolutely mistaken if you think that a casual computer user can know nothing about it, but any good specialist gives you its immediate description, because the current definition of software engineering is still being debated by practitioners today as they struggle to come up with ways to produce software that is "cheaper, better and faster". Cost reduction has been a primary focus of the IT industry since the 1990s. Total cost of ownership represents the costs of more than just acquisition. It includes things like productivity impediments, upkeep efforts and resources needed to support infrastructure.

In Wikipedia, well-known free encyclopedia, we can find such a definition: Software Engineering (SE) is a profession dedicated to