NAME; KENSEIMO ANDY CHRISTOPHER

MATRIC NUMBER; 15/ENG06/042

DEPARTMENT; MECHANICAL ENGINEERING

MEE 586 ASSIGNMENT

ANSWERS:

1. WHAT IS AN INTEGRATED CAD/ CAM?

INTEGRATED CAD/CAM; Integrated CAD/CAM system provides one model supporting both design and manufacturing functions instead of having various file formats, numerous data translations/conversions, and different CAD and CAM models. It’s helpful to contrast the workflow processes associated with the traditional, sequential, non-integrated approach to design through manufacturing versus the concurrent, collaborative workflow process made possible by an integrated CAD/CAM platform. Each approach will be examined in detail for both injection-molded and machined parts.

1. Product cycle to describe the scope of CAD/CAM in the operation of manufacturing firm.



1. Explain seven (7) characteristics of a good CAD software
2. Efficiency; An Efficient software is that which can use less resources such as CPU in terms of time and usage to give a better output.
3. Flexibility; The software must be able to incorporate the design modification without much of difficulty.
4. Maintainability; is the ease with which changes can be made to satisfy new requirements or to correct deficiencies. Well designed software should be flexible enough to accommodate future changes that will be needed as new requirements come to light. Since maintenance accounts for nearly 70% of the cost of the software life cycle [Schach 1999], the importance of this quality characteristic cannot be overemphasized. Quite often the programmer responsible for writing a section of code is not the one who must maintain it. For this reason, the quality of the software documentation significantly affects the maintainability of the software product.
5. Correctness; is the degree with which software adheres to its specified requirements. At the start of the software life cycle, the requirements for the software are determined and formalized in the requirements specification document. Well designed software should meet all the stated requirements. While it might seem obvious that software should be correct, the reality is that this characteristic is one of the hardest to assess. Because of the tremendous complexity of software products, it is impossible to perform exhaustive execution-based testing to insure that no errors will occur when the software is run. Also, it is important to remember that some products of the software life cycle such as the design specification cannot be "executed" for testing. Instead, these products must be tested with various other techniques such as formal proofs, inspections, and walkthroughs.
6. Reusability; is the ease with which software can be reused in developing other software. By reusing existing software, developers can create more complex software in a shorter amount of time. Reuse is already a common technique employed in other engineering disciplines. For example, when a house is constructed, the trusses which support the roof are typically purchased preassembled. Unless a special design is needed, the architect will not bother to design a new truss for the house. Instead, he or she will simply reuse an existing design that has proven itself to be reliable. In much the same way, software can be designed to accommodate reuse in many situations. A simple example of software reuse could be the development of an efficient sorting routine that can be incorporated in many future applications.
7. Reliability; is the frequency and criticality of software failure, where failure is an unacceptable effect or behavior occurring under permissible operating conditions. The frequency of software failure is measured by the average time between failures. The criticality of software failure is measured by the average time required for repair. Ideally, software engineers want their products to fail as little as possible (i.e., demonstrate high correctness) and be as easy as possible to fix (i.e., demonstrate good maintainability). For some real-time systems such as air traffic control or heart monitors, reliability becomes the most important software quality characteristic. However, it would be difficult to imagine a highly reliable system that did not also demonstrate high correctness and good maintainability.
8. Portability; is the ease with which software can be used on computer configurations other than its current one. Porting software to other computer configurations is important for several reasons. First, "good software products can have a life of 15 years or more, whereas hardware is frequently changed at least every 4 or 5 years. Thus good software can be implemented, over its lifetime, on three or more different hardware configurations". Second, porting software to a new computer configuration may be less expensive than developing analogous software from scratch. Third, the sales of "shrink-wrapped software" can be increased because a greater market for the software is available.
9. Explain three (3) divisions of software components
10. Application software is the general designation of computer programs for performing tasks. Application software may be general purpose (word processing, web browsers, etc.) or have a specific purpose (accounting, truck scheduling, etc.). Application software contrasts with system software.
11. System software is a generic term referring to the computer programs used to start and run computer systems including diverse application software and networks.
12. Computer programming tools, such as compilers and linker, are used to translate and combine computer program source code and libraries into executable RAMs (programs that will belong to one of the three said).