COMPUTER PROGRAMMING

Computer programs are developed by computer programmers of software engineers. Computer programming encompasses a broad set of activities that include planning, coding, testing, and documenting. Most programmers participate in all of these phase of program development, but focus on the coding process. Software engineers fend to on designing and testing activities. The programming process begins with a problem statement that helps you clearly define the purpose of a computer program. In the context of programming, a problem statement define certain elements that must be manipulated to achieve a result of goal. A good problem statement for a computer program has three characteristics.

-It specifies any assumptions that define the scope of the problem.

-It clearly specifies the known inform.

-It specifies when the problem has been solved.

In a problem statement, an assumption is something you accept as the in order to processed with program planning. A predictive methodology requires extensive planning and documentation up front. In contrast to predictive methodologies, an a gile methodology focuses on flexible development and specifications that evolve as a project progresses. The phrase programming paradigm refers to a way of conceptualizing and structuring the phases a computer performs. Today’s most popular programming paradigms are: event-driven paradigm, proce dural, object-oriented paradigm, and declarative. The process of coding a computer program depends on the programming language you use and the programming tools you select. Programming languages can be devided into two major categories. Low-level languages and high level languages. Programmers can use a text editor , program editor, or VDE to code computer programs. Visual development environment provides programmers with tools to build substantial sections of a program by pointing and clicking rather that typing lines of code. When a program doesn’t work correctly, it’s usually the result of a syntax, logic or runtime error. Anyone who uses computers is familiar with program documentation in the form of user manuals and help fiels. The term “ratification intelligence” dates back to 1956 and belong to a Stanford researcher John McCarthy, who coinfed the term and definded the key mission of AI as a sub-field of computer science. Basically, artificial intelligence is the ability of a machine or a computer program to think and learn. The concept of AI is based on the idea of building machines capable of thinking, acting, and learning like humans. There are certain things a machine computer program must be capable of to be considered AI. First, it should be able to mimic human thought process and behavior. Second, it should act in a human-like way-intelligent, rational, and ethical. Artificial intelligence, in its turn, is a bunch of technologies that include machine learning and some other technologies like natural language processing, inference algorithms, neural networks.

Hackers, crackers, cybercriminals, or black hats no matter what you call them, their goal is to gain unauthorized access to information. The first line of define in cyber security is computer programmers who create the applications an utilities that you use every day. Some of the most common software security defects include buffer overflows and verbose error messages. A buffer overflow is a condition in which data in memory exceeds expected boundaries and flows into memory areas intended for use by order data. A technique called address space randomization arranges key data areas in locations that are difficult for hackers to predict and target. If those verbose error messages remain when the software ships, they can present attackers with information about the directory location of programs or files. Software security begins when program specifications are formulated. Formal methods help programmers apply logical and mathematical models to software design coding, testing, and verification. Threat modeling is a technique that can be used to identify potential vunerabiliting. There are can be categorized using a mode like STRIDE: spooking, repudiation, service. Software designers can rank threats using the DREAD categories: damage, reproduce, exploit, affected, discovered. Defenessive programming is an approach to software development in which programmers anticipate what might go wrong as their programs run, and make steps to smothyly handle those situations. Techniques associated with defensive programming include source code walkthroughs that can identify security holes; simplification or complex section of code.