The reuse of domain knowledge is clearly a feature that will take different forms in particular domains. Consider what particular forms of experience might be useful in the domains of

 (a) processing financial transactions  
 (b) real-time process control for chemical plants  
 (c) writing compilers for imperative programming languages.

In contrast to broad (or domain-independent) knowledge, domain knowledge is knowledge of a particular, specialised discipline or field. The phrase is frequently used to allude to a broader discipline, such as when characterising a software engineer who has both general knowledge of computer programming and industry-specific domain expertise. Domain-savvy individuals are frequently considered as specialists or authorities in their industry.

1. processing financial transactions

step 1:

Transaction processing is the broad category of routine tasks essential to the accounting and financial management of every business. Sales, purchases, receipts, and payments are the four primary categories of business financial transactions.

Transaction processing generates a significant number of accounting entries and financial oversight, as we briefly mentioned before. Financial controls are required at multiple stages of each transaction process and across numerous business units, which makes the financial ecosystem more complex.

Step 2:

Innovation and knowledge management have always been handled separately in financial service businesses. Financial services knowledge management is now more equipped than ever to provide value to commercial and investment banks, insurance companies, credit unions, brokerage firms, wealth management advice organisations, and others after they started to merge two years ago.

A knowledge management system in the financial services industry makes it possible to create processes and makes connections between various operations. Additionally, it makes it possible to provide knowledge to those in need at the right time, swiftly and simply.

In the banking sector and other industries, knowledge management can also increase competitiveness, safeguard financial services providers from financial risk, reduce operational inefficiency, and streamline internal communications.

Information management in the financial services industry is a challenging task. Software for knowledge management makes the process simpler, allowing firms to run more effectively and satisfactorily satisfy consumer expectations.

Here are several ways knowledge management keeps businesses safer and aids them in their digital transformations, in addition to assisting banks and other financial institutions to minimise or avoid crises.

1. real-time process control for chemical plants

step 1:

Factory managers can closely monitor the levels of energy efficiency of production processes thanks to process control. Less defective items will be generated if the machinery is operated at its best capacity without wasting energy needlessly.

Step 2:

If we have enough knowledge in real-time process control for chemical plants, that will

aids in problem identification in a chemical process using data from plant information systems. Principal component analysis (PCA) techniques of multivariate statistical analysis were employed in this work to find faults. A collection of variable combinations, or the identification of major components, which show the tendency of variables and operating data, are the basic ideas. In addition to traditional statistical process control, PCA can decrease the dimension of variables using process monitoring. As a result, they are recognised as appropriate techniques for handling large data sets with numerous dimensions.

1. writing compilers for imperative programming languages.

Step 1:

Software that uses imperative programming use statements that alter the state of the programme. An imperative programme includes of instructions for the computer to follow, much as how the imperative mood communicates directives in spoken languages. In contrast to high-level explanations of a program's anticipated outcomes, imperative programming focuses on detailing how a programme works step by step.

Step 2:

The compiler writer has two options for optimising the imperative representation stages of their compiler: they can implement instruction scheduling or combine multiplication and division into a single operation. If they don't utilise a code generating library like LLVM, they will have to manage register and stack slot allocation. On this type of data structure, many of these optimizations can be challenging, but some (register allocation) must be carried out on an imperitive style data structure.

The compiler developer has the option to optimise in the higher-level functional form as well. This is typically much simpler. Inlining is an excellent illustration of this; in the functional form, inlining can be used by substituting. While it can be much more tiresome at the essential level,