# Description :-

Our Expert System is aimed at solving the challenge of deciding the appropriate programming language for a user based on a set of queries depicting the aspirations, likings and preferences of the user. This system asks a myriad of questions - ranging from the platform/operating system , implementation goals, monetary goals of the user - fed a priori in the knowledge base. The decision for the appropriate programming language is taken based on the answers to these questions fed by the user.

# How our Expert System works :-

Our Expert System is a rule based Expert System written in prolog. The execution of Prolog is done via depth - first backward chaining. Backward chaining (or backward reasoning) is an inference method that can be described (in lay terms) as working backward from the goal(s). It is used in automated theorem provers, inference engines, proof assistants and other artificial intelligence applications.

Backward chaining starts with a list of goals (or a hypothesis) and works backwards from the consequent to the antecedent to see if there is data available that will support any of these consequents. An inference engine using backward chaining would search the inference rules until it finds one which has a consequent (Then clause) that matches a desired goal. If the antecedent (If clause) of that rule is not known to be true, then it is added to the list of goals (in order for one's goal to be confirmed one must also provide data that confirms this new rule).

Our program is divided into the following parts (with functions in each part):-

• Introduction – main, welcome

• Knowledge Base – language

• Inference Engine – findLanguage, why, preferredPlatform,

preferredMobileOS, web, microsoftOpinion,

workFor, tryNew, favouriteToy,

learningPreference, whichCar, ask

• User Interface – answers, question, answer, ask, parse