CS 5346: Adv Artificial Intelligence

EXPERT SYSTEM FOR CAREER ADVISING

MEGHANA BODDULURI[A04898234]

Table of Contents

ABSTRACT

The aim of this project is to build an expert system which provides guidance to students to purse their courses in their specified interest field and mention the professions available for their respective courses.

Understanding, analyzing and using the data to arrive at some conclusion play vital role in creating efficient Artificial Intelligence. Inference engine allows the use of inference rule for deriving a conclusion from the already known data, which in our project is referenced as Knowledge base.

This project is a study work, java implementation and the survey for better understanding of the two-interesting principle of inference engine in expert system, namely Backward chaining and Forward chaining, using these two methodologies and implement working example using Object oriented programming and generate a report with the inputs from 5 students.

INTRODUCTION

Advancement of the technologies and the research in different areas have increased the scope of new subjects and courses as well as professions related to the respective courses. But unfortunately, the new courses are not being highlighted to students cause of lack of information about the available courses in their interested areas of study.

The need of Advisors for college Freshmen in guiding them with courses can be automated using Expert system which suggests the individual with the courses available in their desired profession.

FORMAL DESCRIPTION OF THE PROBLEM

Create an intelligent computer expert system for career guidance of students to purse courses. After considering their interest and determining the profession, the system should display recommended courses which help to obtain that profession. Perform research using Web or any other source to collect knowledge about the research areas and different professions in demand and create a knowledge base. The student will feed their interested areas of studies. The expert system will analyze the job domain of the individual and will recommend the subjects. After collecting knowledge, develop a decision tree. Then transform the decision tree into rules. The profession Diagnosis decision tree should be big enough to generate a minimum of ten professions. The courses analysis decision tree should provide at least five courses for desired profession. Rules should contain variables.

Implement the expert system program, employing Backward Chaining and Forward Chaining methodologies. The output generated as profession by backward chaining will be provided as input for forward chaining.

SOLUTION OF THE PROBLEM

Backward Chaining methodology would be used to determine the profession in which the student is interested, professions in Engineering, Science, Business, Medical, English, Geography, Psychology, Agriculture, Health Care, and Education.

Forward Chaining methodology would then be used to recommend a specific area in which a student should pursue. For example, if the profession is medical, the area could be chosen from mental health, oncologist, orthopedic, cardiologist, and General practitioner etc.; if the profession is health care, the area could be chosen from nursing, Health Services Management, Healthcare Administration, Physician Assistant, and Public Health, etc.

GOALS OF THE PROJECT

- At least 10 professions and 5 specific areas of coursework from each profession must be considered while establishing the knowledge base and its respective decision tree.
- employing Software Engineering principles which prohibits 'GO TO' statements and discourage global variable.
- Separate Knowledge base and Inference Engine parts of each program and bring efficiency in functionality and output.
- Efficiency methods include dynamic memory management, use of objects, and Hashing functions.
- 'main' function will call the two other functions, i.e., 'BC_Functionality' and 'FC_Functionality'.
- write an error-free program in JAVA using OOPS concepts.

INPUTS FOR THE PROJECT

- Knowledge Base: Establish Knowledge based and feed it to the System. The knowledge base can be established from various sources online.
- Input from the user: Develop user-friendly interface which receives the input in restricted English format.

EXPECTED OUTPUT OF THE PROJECT

- Profession of the individual determined by backward chaining.
- Specialization area for that profession to be determined by forward chaining.
- On interest of user, display knowledge base and clause variable list.

TEAM MEMBERS

- Divya Viswanathan(A04859120)
- Meghana Bodduluri(A04898234)

CONTRIBUTION

- ***Knowledge Base(Decision tree and If-Then rules)*:** I created the knowledge base for backward chaining by a decision tree and helped in drafting forward chaining by browsing web knowledge and maintaining the project requirements, while Divya completed the forward chaining rules and decision tree. Both of us got together and finalized Decision Tree for FC and BC by eliminating the ambiguities.

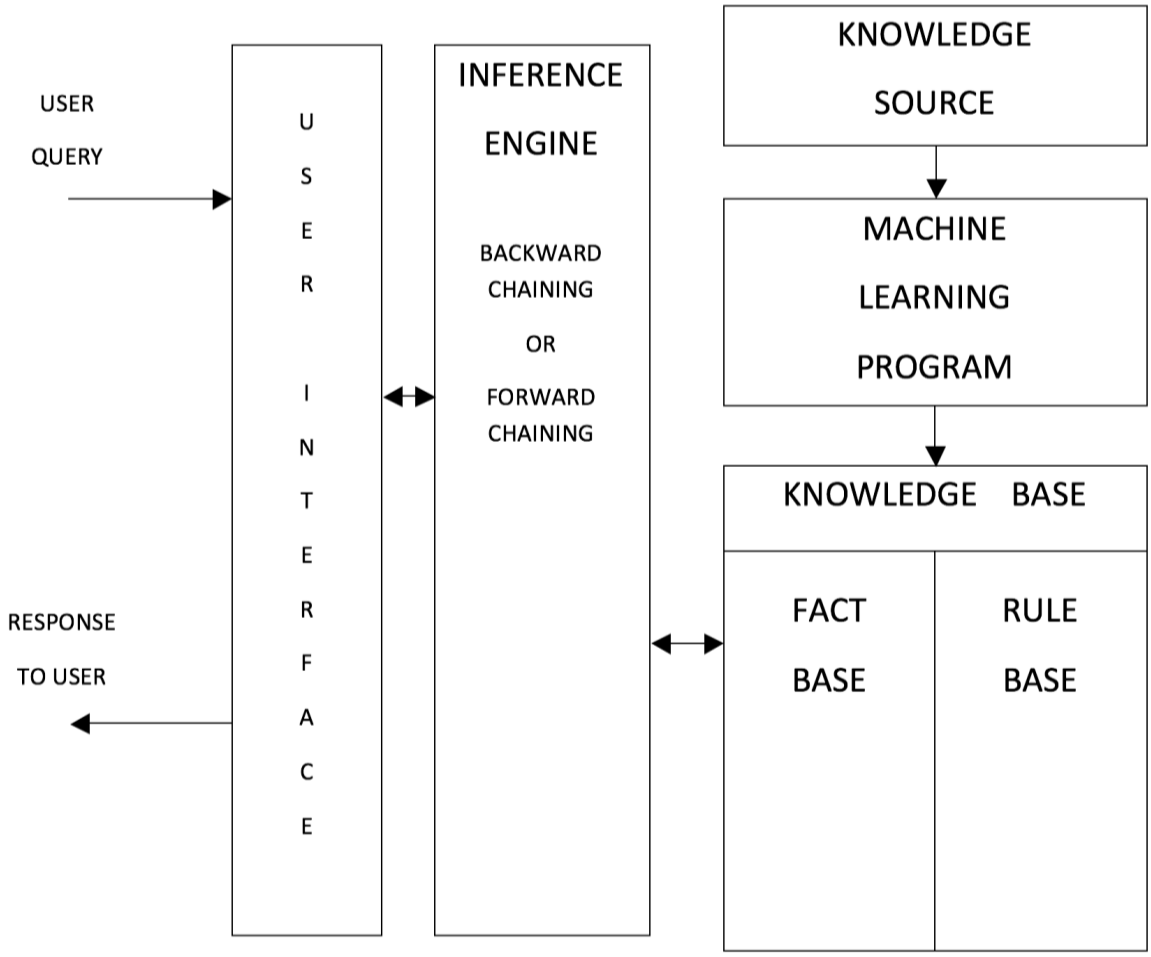
- ***Program:*** I worked with implementation of Backward Chaining function, while Divya worked in the implementation of Forward Chaining function. We then worked on integration of the two modules together. We then, reviewed each-others code in terms of following: -
  - Following the coding standards
  - Ensuring there are intermediate appropriate yet meaningful output statements
  - Ensuring there are in-line meaningful comments for each and every function/logic defined in the program.
  - Ensuring it is Optimized to the best of our knowledge.

EXPERT SYSTEM

The expert systems are the computer applications developed to solve complex problems in a particular domain, emulates the decision-making ability of a human expert. The expert systems are capable of −

- Advising
- Instructing and assisting human in decision making
- Demonstrating
- Deriving a solution
- Diagnosing
- Explaining
- Interpreting input
- Predicting results
- Justifying the conclusion
- Suggesting alternative options to a problem

SOFTWARE ARCHITECTURE



COMPONENTS OF AN EXPERT SYSTEM

The components of the expert system consist of **4** major parts. They are – **User Interface, Inference Engine & Knowledge Base**.

User Interface:

- It enables the users to enter instruction and information into the expert system and to receive information from it.
- user can use method for input command, natural language and customize the interface.

Inference Engine:

- The inference engine is one of the most important components of an expert system, it is the rule that defines how the expert process interprets the knowledge in an appropriate manner.
- The inference engine work in either forward chaining or backward chaining.

Knowledge Base:

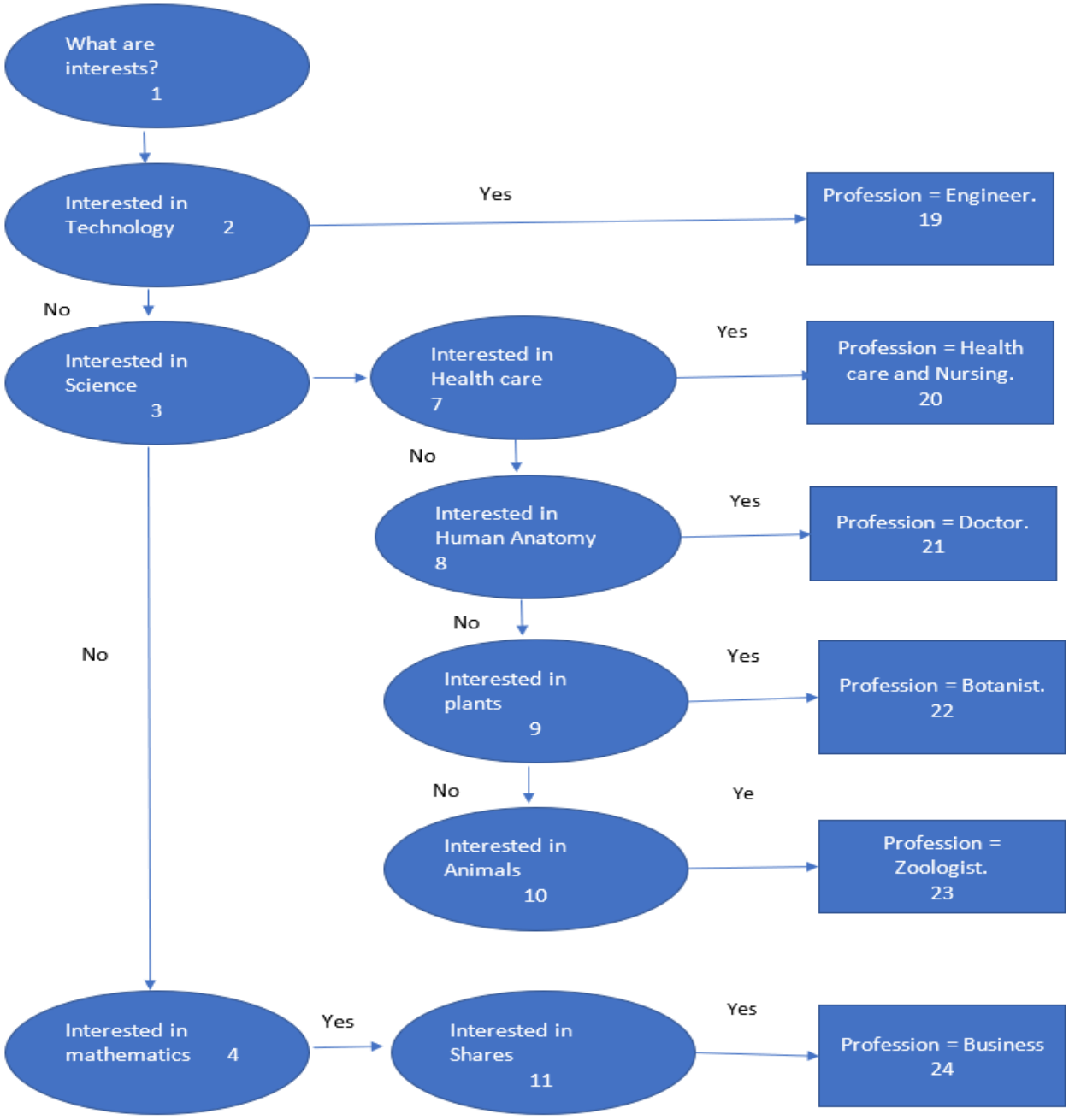
- It contains the fact that describes the problem area and knowledge representation technique that describes the methodology for inference engine.
- The knowledge of human experts is translated into the "if-then" statements.

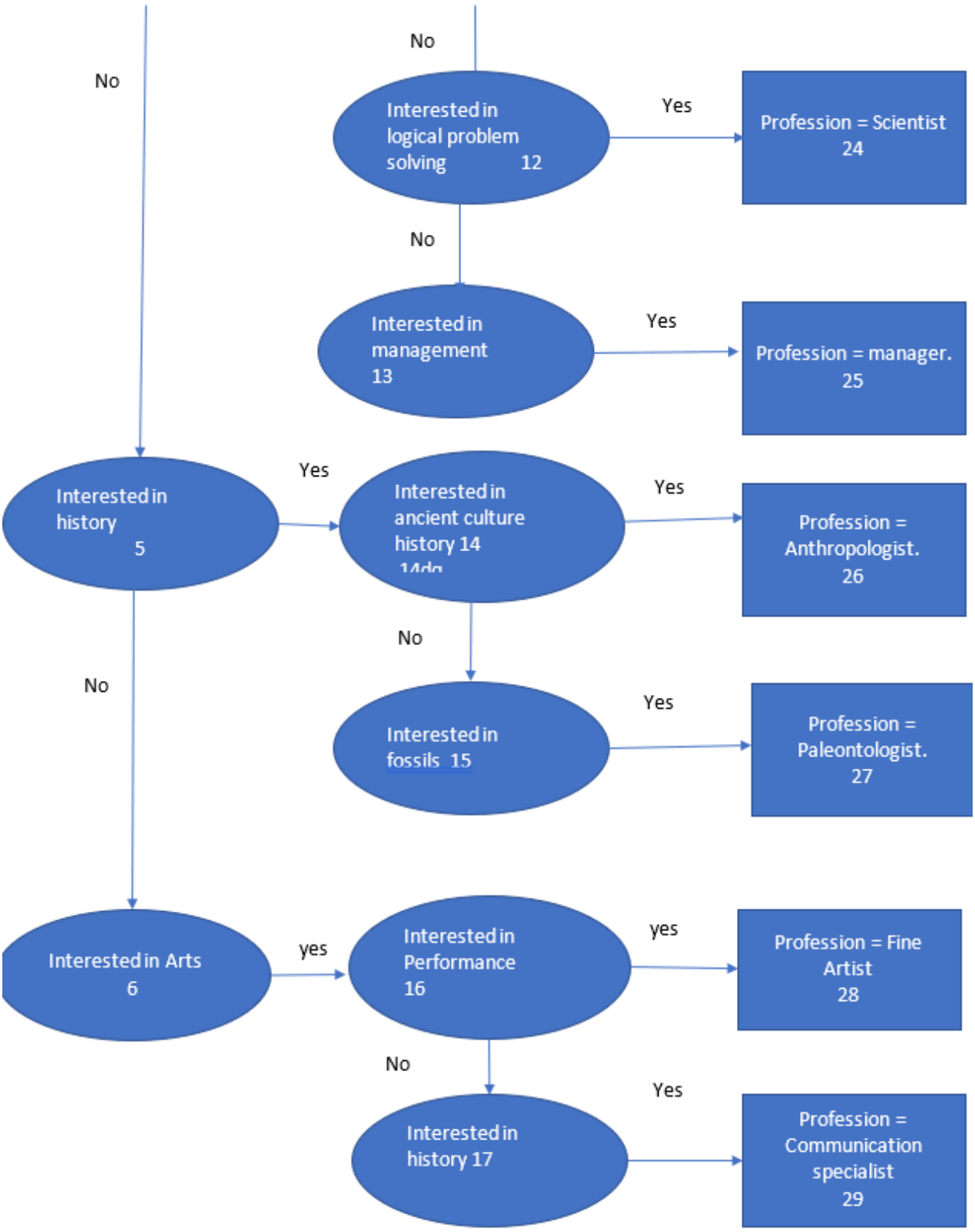
METHODOLOGIES OF AN INFERENCE ENGINE

BACKWARD CHAINING

- Backward chaining is an inference method that is a goal driven process and top-down form of reasoning search strategy by starting from a conclusion, result or goal and going backward to infer the conditions from which it resulted
- In the rules table, it seeks out any actions that are specified in *if-then* statements, applying logic to determine which of the possible actions would have caused the result.

DECISION TREE




No
No
Interested in logical problem solving 12
Yes
Profession = Scientist 24
No
Interested in management 13
Yes
Profession = manager. 25
Interested in history 5
Yes
Interested in ancient culture history 14
Yes
Profession = Anthropologist. 26
No
No
Interested in fossils 15
Yes
Profession = Paleontologist. 27
Interested in Arts 6
yes
Interested in Performance 16
yes
Profession = Fine Artist 28
No
Interested in history 17
Yes
Profession = Communication specialist 29

IF-THEN RULES

Rule No	Rule	Path
10	If INTEREST = YES And TECHNOLOGY = YES Then PROFESSION = Engineer	1,2,19
20	If INTEREST = YES And SCIENCE = YES And HEALTHCARE = YES Then PROFESSION = Nursing and Healthcare	1,2,3,7,20
30	If SCIENCE = YES And HEALTHCARE = NO And HUMAN_ANOTOMY= YES Then PROFESSION = Doctor.	1,2,3,7,8,21
40	If SCIENCE = YES And HEALTHCARE = NO And HUMAN_ANOTOMY = NO And PLANTS= YES Then PROFESSION = Botanist.	1,2,3,7,8,9,22
50	If SCIENCE = YES And HEALTHCARE = NO And HUMAN_ANOTOMY = NO And PLANTS= NO And ANIMALS = YES Then PROFESSION = Zoologist	1,2,3,7,8,9,10,23
60	If MATHS = YES And SHARES = YES Then PROFESSION = Business.	1,2,3,4 ,11,24
70	If MATHS = YES And SHARES = NO And LOGICAL = YES Then PROFESSION = Scientist	1,2,3,4,11,12,24
80	If MATHS = YES And SHARES = NO And LOGICAL = NO And MANAGEMENT =YES Then PROFESSION = Manager.	1,2,3,4,11,12,13,25
90	If HISTORY = YES And CULTURE_HISTORY = YES Then PROFESSION =Anthropologist.	1,2,3,4,5,14,26
100	If HISTORY = YES And CULTURE_HISTORY = NO And FOSSILS = YES Then PROFESSION =Paleontologist.	1,2,3,4,5,15,27

110	If ARTS = YES And MUSIC = YES Then PROFESSION = Fine_Artist.	1,2,3,4,5,6,17,28
120	If ARTS = YES And MUSIC = NO And LANGUAGE = YES Then PROFESSION = Communication specialist.	1,2,3,4,5,6,17,18,29

DATA STRUCTURES USED IN BACKWARD CHAINING

CLAUSE VARIABLE LIST

1	Interest
2	Technology
3	
4	
5	
6	
7	Interest
8	Science
9	HealthCare
10	
11	
12	
13	Science
14	HealthCare
15	Human Anatomy
16	
17	
18	
19	Science
20	HealthCare
21	Human Anatomy
22	Plants
23	
24	
25	Science
26	HealthCare
27	Human Anatomy
28	Plants
29	Animals
30	
31	Interest
32	Maths
33	Shares

34	
35	
36	
37	Maths
38	Shares
39	Logical
40	
41	
42	
43	Maths
44	Shares
45	Logical
46	
47	
48	
49	Maths
50	Shares
51	Logical
52	Management
53	
54	
55	Interest
56	History
57	Culture history
58	
59	
60	
61	History
62	culture_history
63	Fossil
64	
65	
66	
67	Interest
68	Arts
69	Music
70	
71	
72	
73	Arts
74	Music
75	Language

VARIABLE LIST

1.	Interest	NI
2.	Profession	NI
3.	Technology	NI
4.	Science	NI
5.	HealthCare	NI
6.	Human Anatomy	NI
7.	Plants	NI
8.	Animals	NI
9.	Math's	NI
10.	Shares	NI
11.	Logical	NI
12.	Management	NI
13.	History	NI
14.	Culture History	NI
15.	Fossil	NI
16.	Arts	NI
17.	Performance	NI
18.	Language	NI

CONCLUSION VARIABLE LIST

10=Profession
20=Profession
30=Profession
40=Profession
50=Profession
60=Profession
70=Profession
80=Profession
90=Profession
100=Profession
110=Profession
120=Profession

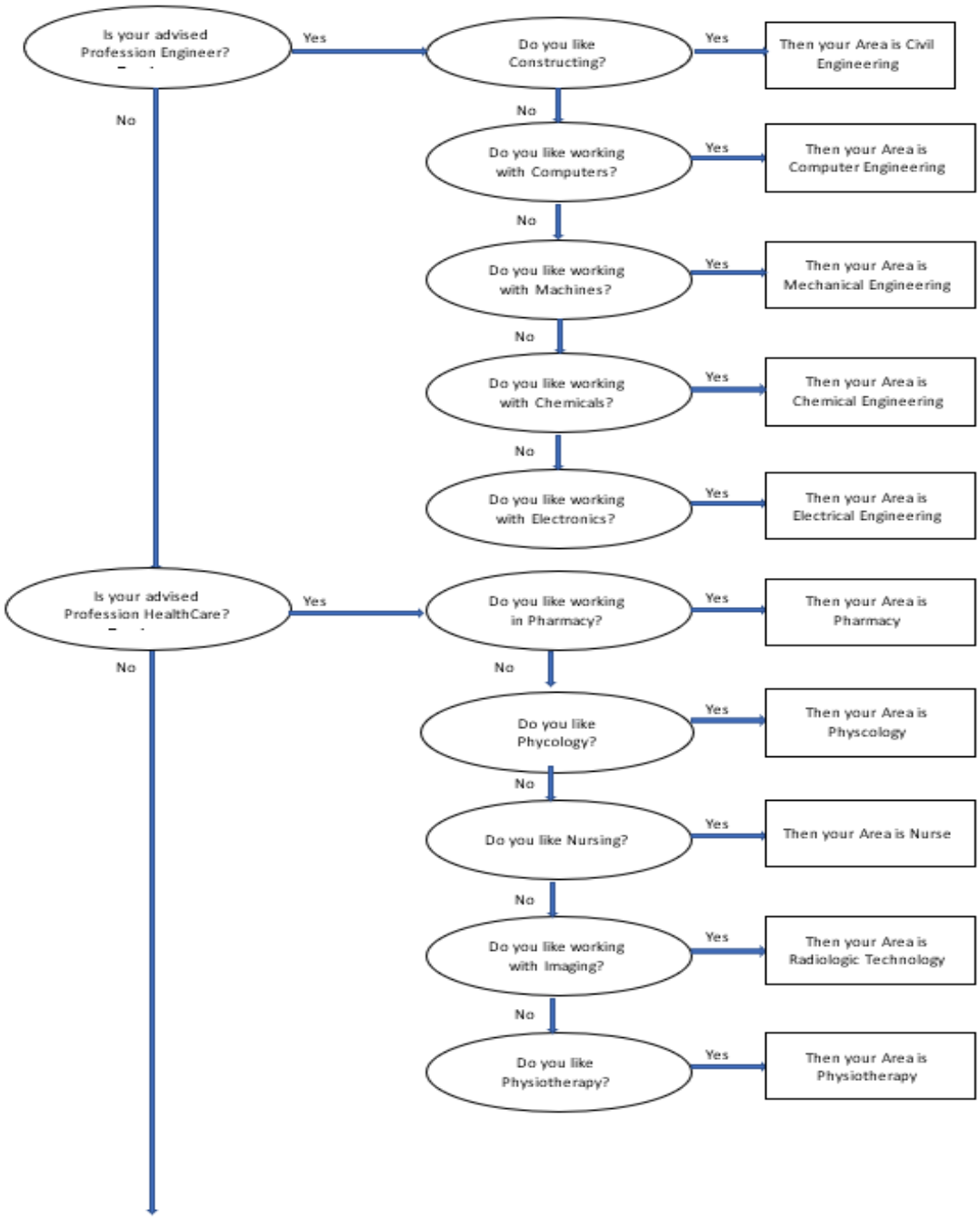
Steps for implementing backward chaining

1. Create an indexed knowledge base in the format of if -then conditions with relation to decision tree.
2. Load all the variables of If part of rule to **Variable list.**
3. Create a **clause variable list** for all rules by using Clause_no = 4*((RuleNo/10)-1) +1.
   It is 4 in this case because, this is a system that reserves 4 locations for each rule.
4. Load all the conclusion variables of then part of rules into **Conclusion list**.
5. Determine the conclusion variable in conclusion list.
6. Search the Conclusion List for the first instance of the conclusion derived in step 4. If found, place the rule number and the calculated clause no in the **Conclusion Stack.**
7. Instantiate the IF clause (each condition variable) that corresponds to the rule number found in step 6.
8. If the variable in IF is not instantiated and if the variable is not present in Conclusion List, ask the user for the same with appropriate prompts.
9. If one of the clauses is in the Conclusion List, then place the corresponding rule number and its calculated clause number over the top of the Conclusion Stack and go back to step 6.
10. If the statements on the top of the stack cannot be executed then remove that rule from the top of the stack and search the conclusion list for another instance of that conclusion variable's name. If found, go back to step 6.
11. Once the IF is executed successfully, thus leading to instantiation of a conclusion variable, remove the corresponding rule number/clause number from top of the stack.
12. Repeat from step 6 for the rule number/clause number currently present in top of the stack.
13. Repeat until there are no more entries in the conclusion stack.

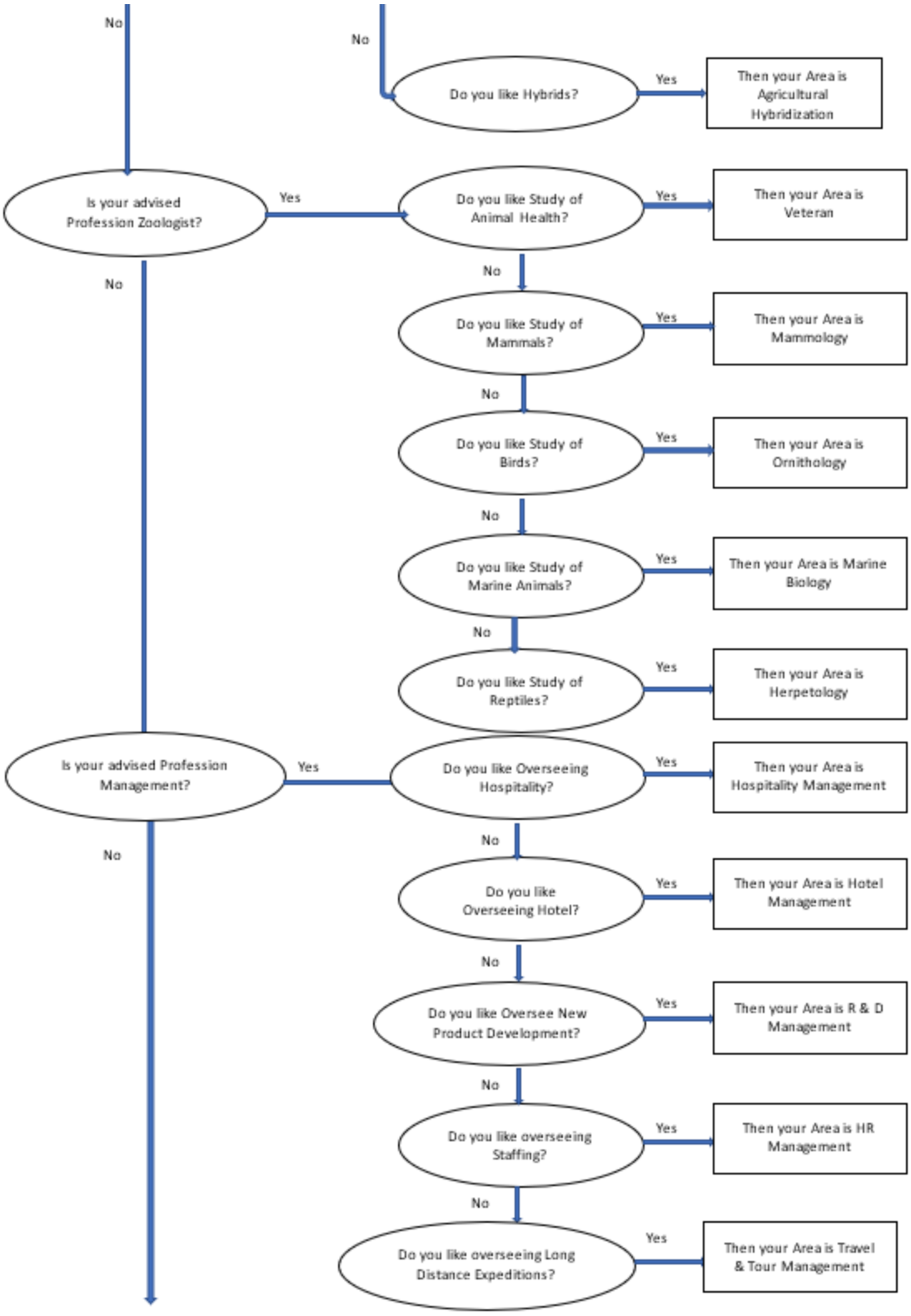
FORWARD CHAINING

- Forward chaining is an inference method that is a data-driven process by executing available data and uses inference rules to extract more data (from an end user) until a goal is reached. An inference engine using forward chaining searches the inference rules until it finds one where the antecedent (If clause) is known to be true.
- It starts with some facts and applies rules to find all possible conclusions

DECISION TREE




Is your advised Profession Doctor?
Yes
No
Do you like Study of Heart?
Yes
Then your Area is Cardiology
No
Do you like Study of Teeth?
Yes
Then your Area is Dental
No
Do you like Study of Bones?
Yes
Then your Area is Orthopedic
No
Do you like Study of Nerves?
Yes
Then your Area is Neurology
No
Do you like working with ChildrenHealth?
Yes
Then your Area is Pediatric
Is your advised Profession Botanist?
Yes
No
Do you like Growing Plants?
Yes
No
Do you like Growing them in Small Scale?
Yes
Then your Area is Horticulture
No
Do you like Growing them in Large Scale?
Yes
Then your Area is Agriculture
No
Do you like Study of Plants?
Yes
Do you like Plant Internal Structure?
Yes
Then your Area is Plant Anatomology
No
Do you like Plant External Structure?
Yes
Then your Area is Plant Morphology


No
No
Do you like Hybrids?
Yes
Then your Area is Agricultural Hybridization
Is your advised Profession Zoologist?
Yes
Do you like Study of Animal Health?
Yes
Then your Area is Veteran
No
No
Do you like Study of Mammals?
Yes
Then your Area is Mammology
No
Do you like Study of Birds?
Yes
Then your Area is Ornithology
No
Do you like Study of Marine Animals?
Yes
Then your Area is Marine Biology
No
Do you like Study of Reptiles?
Yes
Then your Area is Herpetology
Is your advised Profession Management?
Yes
Do you like Overseeing Hospitality?
Yes
Then your Area is Hospitality Management
No
No
Do you like Overseeing Hotel?
Yes
Then your Area is Hotel Management
No
Do you like Oversee New Product Development?
Yes
Then your Area is R & D Management
No
Do you like overseeing Staffing?
Yes
Then your Area is HR Management
No
Do you like overseeing Long Distance Expeditions?
Yes
Then your Area is Travel & Tour Management


Is your advised Profession Business?
Yes
No
Do you like Finance?
Yes
Then your Area is Accountant
No
Do you like Sales?
Yes
Then your Area is Marketing
No
Do you like Manufacturing Products?
Yes
Then your Area is Manufacturing Business
No
Do you like Cross-Border Trade?
Yes
Then your Area is International Business
No
Do you like Business Law?
Yes
Then your Area is Business Law
Is your advised Profession Paleontologist?
Yes
No
Do you like Study of Microscopic Fossils?
Yes
Then your Area is Microscopic Paleontology
No
Do you like Study of Vertebrate Animals?
Yes
Then your Area is Vertebrate Paleontology
No
Do you like Study of Fossilized Invertebrate Animals?
Yes
Then your Area is Invertebrate Paleontology
Yes
Do you like Study of Fossilized Plants?
Yes
Then your Area is Paleo botany
Yes
Do you like Study of Fossil Preservation?
Yes
Then your Area is Taphonomology


Is advised Profession Anthropologist?
Yes
Do you like Study of Pre-Historic Material Culture??
Yes
Then your Area is Archaeology
No
Do you like Study of Origins of Languages?
Yes
Then your Area is Linguistic Anthropology
No
Do you like Study of Ancient Human Society?
Yes
Then your Area is Cultural Anthropology
No
Do you like Study of Human Evolution
Yes
Then your Area is Paleoanthropology
No
Do you like Study of Human Remains & DNA
Yes
Then your Area is Forensic Anthropology
No
Is your advised Profession Fine-Artists?
Yes
Do you like Study of Music?
Yes
Then your Area is Musician
No
Do you like Study of Carving?
Yes
Then your Area is Sculpture
No
Do you like Study of Dance?
Yes
Then your Area is Dance
No
Do you like Cinematography?
Yes
Then your Area is Films & Media
No
Do you like Capturing Moments or Scenery
Yes
Then your Area is Photography
No


Is your advised Profession Communication Specialist?
Yes
Do you like Build, Maintain & Manage Reputation for Client?
Yes
Then your Area is Public Relations
No
Do you like Promote Product Creatively?
Yes
Then your Area is Advertising
No
Do you like Writing Broadcast News?
Yes
Then your Area is Journalism
No
Do you like learn Multiple Languages & Cultures?
Yes
Then your Area is International Relations
No
Do you like Interact & teach Students?
Yes
Then your Area is Professor
No
Is your advised Profession Scientist?
Yes
Do you like Scientific Study ~~of~~ Universe
Yes
Then your Area is Cosmology
No
Do you like Scientific Study of Nuclear Process
Yes
Then your Area is Nuclear Science
No
Do you like Scientific Study of Plasma?
Yes
Then your Area is Plasma Science
No
Do you like Scientific study of Rocket Science?
Yes
Then your Area is Rocket Scientist
No
Do you like Scientific study of Quantum Science?
Yes
Then your Area is Quantum Physics

IF-THEN RULES

	Rules
10	IF Profession = Engineer and SubInterest=Construction THEN Area=Civil_Engineering
20	IF Profession = Engineer and SubInterest=Computers THEN Area=Computer_Engineering
30	IF Profession = Engineer and SubInterest=Chemicals THEN Area=Chemical_Engineering
40	IF Profession = Engineer and SubInterest=Machines THEN Area=Mechanical_Engineering
50	IF Profession = Engineer and SubInterest=Electronics THEN Area=Electrical_Engineering
60	IF Profession = HealthCare and SubInterest=medicine_drugs THEN Area=Pharmacy
70	IF Profession = HealthCare and SubInterest=mental health THEN Area=Psychology
80	IF Profession = HealthCare and SubInterest=Nursing THEN Area=Nurse
90	IF Profession = HealthCare and SubInterest=Imaging THEN Area=Radiologic_Technology
100	IF Profession = HealthCare and SubInterest=Physiotherapy THEN Area=Physiotherapy
110	IF Profession = Doctor and SubInterest=Heart THEN Area=Cardiology

120 IF Profession = Doctor and SubInterest=Teeth
THEN Area=Dental

130 IF Profession = Doctor and SubInterest=Bones
THEN Area=Orthopedics

140 IF Profession = Doctor and SubInterest=Nerves
THEN Area=Neurology

150 IF Profession = Doctor and SubInterest=Children_General_Health
THEN Area=Pediatrician

160 IF Profession = Botanist and SubInterest=Grow_Plants and Scale=Small
THEN Area=Horticulture

170 IF Profession = Botanist and SubInterest=Grow_Plants and Scale=Large
THEN Area=Agriculture

180 IF Profession = Botanist and SubInterest=Study_Plants and Focus=Internal_Study
THEN Area=Plant_Anatomy

190 IF Profession = Botanist and SubInterest=Study_Plants and Focus=External_Study
THEN Area=Plant_Morphlogy

200 IF Profession = Botanist and SubInterest=Plant_Hybrids
THEN Area=Agricultural_Hybridization

210 IF Profession = Botanist and SubInterest=Diseased_Plant
THEN Area=Plant_Pathology

220 IF Profession = Zoologist and SubInterest=Animal_Health
THEN Area=Animal_Veteran

230 IF Profession = Zoologist and SubInterest=Study_Mammals
THEN Area=Mammology

240 IF Profession = Zoologist and SubInterest=Study_Birds
THEN Area=Ornitholgy

250 IF Profession = Zoologist and SubInterest=Marine_Animals
THEN Area=Marine_Biology

260 IF Profession = Zoologist and SubInterest=Study_Reptiles
THEN Area=Herpetology

270 IF Profession = Management and SubInterest=Oversee_Hospitality
THEN Area=Hospitality_Management

280 IF Profession = Management and SubInterest=Oversee_Hotel
THEN Area=Hotel_Management

290 IF Profession = Management and SubInterest=Oversee_New_Product_Development
THEN Area=R and D_Management

300 IF Profession = Management and SubInterest=Oversee_Staffing
THEN Area=HR_Management

310 IF Profession = Management and SubInterest=Oversee_Long_Distance_Expedition
THEN Area=TravelandTour_Management

320 IF Profession = Management and SubInterest=Oversee_Budget
THEN Area=Budget_Management

330 IF Profession = Business and SubInterest=Finance
THEN Area=Accountant

340 IF Profession = Business and SubInterest=Sales
THEN Area=Marketing

350 IF Profession = Business and SubInterest=Manufacturing
THEN Area=Manufactuing_Business

360 IF Profession = Business and SubInterest=Merchandise
THEN Area=Merchandising_Business

370 IF Profession = Business and SubInterest=Business_Law
THEN Area=Business_Lawyer

380 IF Profession = Business and SubInterest=Cross_Border_Trade
THEN Area=International_Business

390 IF Profession = Paleontologist and SubInterest=Mircoscopic_Fossils
THEN Area=Micropaleontology

400 IF Profession = Paleontologist and SubInterest=Fossilized_Plants
THEN Area=Paleo_Botany

410 IF Profession = Paleontologist and SubInterest=Fossilized_Animals and Focus=Invertebrate
THEN Area=Invertebrate_Paleontology

420 IF Profession = Paleontologist and SubInterest=Fossilized_Animals and Focus=Vertebrate
THEN Area=Vertebrate_Paleontology

430 IF Profession = Paleontologist and SubInterest=Preserve_Fossils
THEN Area=Taphonomy

440 IF Profession = Anthropologist and SubInterest=Pre-historic_Material_Culture_Study
THEN Area=Archaeology

450 IF Profession = Anthropologist and SubInterest=Language_Origins
THEN Area=Linguistic_Anthropology

460 IF Profession = Anthropologist and SubInterest=Human_Society_Study
THEN Area=Cultural_Anthropology

470 IF Profession = Anthropologist and SubInterest=Study_Human_Evolution
THEN Area=Paeo_Anthropology

480 IF Profession = Anthropologist and SubInterest=Study_Human_RemainsandDNA
THEN Area=Forensic_Anthropology

490 IF Profession = Fine_Artist and SubInterest=Music
THEN Area=Musician

500 IF Profession = Fine_Artist and SubInterest= Carving
THEN Area=Sculpturer

510 IF Profession = Fine_Artist and SubInterest=Dance
THEN Area=Dance

520 IF Profession = Fine_Artist and SubInterest=Cinematography
THEN Area=Film_Media

530 IF Profession = Fine_Artist and SubInterest=Capture_Moments
THEN Area=Photographer

540 IF Profession = Comunication_Specialist and SubInterest=Build_Client_Reputation
THEN Area=Public_Relations

550 IF Profession = Comunication_Specialist and SubInterest=Promote_Product_Creatively
THEN Area=Advertising

560 IF Profession = Comunication_Specialist and SubInterest=Broadcast_News
THEN Area=Journalisms

570 IF Profession = Comunication_Specialist and SubInterest=Learn_Foreign_Lanaguages_Cultures THEN Area=International_Relations

580 IF Profession = Comunication_Specialist and SubInterest=Teach_Interact_Students THEN Area=Professor

590 IF Profession = Scientist & SubInterest=Scientifc_Study_of_Universe THEN Area=Cosmologist

600 IF Profession = Scientist & SubInterest=Scientifc_Study_of_Nuclear_Process THEN Area=Nuclear_Physicist

610 IF Profession = Scientist & SubInterest=Scientifc_Study_of_Plasma THEN Area=Plasma_Physicist

620 IF Profession = Scientist & SubInterest=Scientifc_Study_of_Rocket_Science THEN Area=Rocket_Scientist

630 IF Profession = Scientist & SubInterest=Scientifc_Study_of_Quantum_Science THEN Area=Quantum_Physicist

DATA STRUCTURES USED IN BACKWARD CHAINING

CLAUSE VARIABLE LIST

Clause No	Clause Variable	Clause No	Clause Variable
1.	PROFESSION	117.	PROFESSION
2.	SUBINTEREST	118.	SUBINTEREST
3.		119.	
4.		120.	
5.	PROFESSION	121.	PROFESSION
6.	SUBINTEREST	122.	SUBINTEREST
7.		123.	
8.		124.	
9.	PROFESSION	125.	PROFESSION
10.	SUBINTEREST	126.	SUBINTEREST
11.		127.	
12.		128.	
13.	PROFESSION	129.	PROFESSION
14.	SUBINTEREST	130.	SUBINTEREST

15.		131.	
16.		132.	
17.	PROFESSION	133.	PROFESSION
18.	SUBINTEREST	134.	SUBINTEREST
19.		135.	
20.		136.	
21.	PROFESSION	137.	PROFESSION
22.	SUBINTEREST	138.	SUBINTEREST
23.		139.	
24.		140.	
25.	PROFESSION	141.	PROFESSION
26.	SUBINTEREST	142.	SUBINTEREST
27.		143.	
28.		144.	
29.	PROFESSION	145.	PROFESSION
30.	SUBINTEREST	146.	SUBINTEREST
31.		147.	
32.		148.	
33.	PROFESSION	149.	PROFESSION
34.	SUBINTEREST	150.	SUBINTEREST
35.		151.	
36.		152.	
37.	PROFESSION	153.	PROFESSION
38.	SUBINTEREST	154.	SUBINTEREST
39.		155.	
40.		156.	
41.	PROFESSION	157.	PROFESSION
42.	SUBINTEREST	158.	SUBINTEREST
43.		159.	
44.		160.	
45.	PROFESSION	161.	PROFESSION
46.	SUBINTEREST	162.	SUBINTEREST
47.		163.	
48.		164.	
49.	PROFESSION	165.	PROFESSION
50.	SUBINTEREST	166.	SUBINTEREST
51.		167.	
52.		168.	
53.	PROFESSION	169.	PROFESSION
54.	SUBINTEREST	170.	SUBINTEREST
55.		171.	
56.		172.	

57.	PROFESSION	173.	PROFESSION
58.	SUBINTEREST	174.	SUBINTEREST
59.		175.	
60.		176.	
61.	PROFESSION	177.	PROFESSION
62.	SUBINTEREST	178.	SUBINTEREST
63.	SCALE	179.	SCALE
64.		180.	
65.	PROFESSION	181.	PROFESSION
66.	SUBINTEREST	182.	SUBINTEREST
67.	SCALE	183.	SCALE
68.		184.	
69.	PROFESSION	185.	PROFESSION
70.	SUBINTEREST	186.	SUBINTEREST
71.	FOCUS	187.	FOCUS
72.		188.	
73.	PROFESSION	189.	PROFESSION
74.	SUBINTEREST	190.	SUBINTEREST
75.	FOCUS	191.	FOCUS
76.		192.	
77.	PROFESSION	193.	PROFESSION
78.	SUBINTEREST	194.	SUBINTEREST
79.		195.	
80.		196.	
81.	PROFESSION	197.	PROFESSION
82.	SUBINTEREST	198.	SUBINTEREST
83.		199.	
84.		200.	
85.	PROFESSION	201.	PROFESSION
86.	SUBINTEREST	202.	SUBINTEREST
87.		203.	
88.		204.	
89.	PROFESSION	205.	PROFESSION
90.	SUBINTEREST	206.	SUBINTEREST
91.		207.	
92.		208.	
93.	PROFESSION	209.	PROFESSION
94.	SUBINTEREST	210.	SUBINTEREST
95.		211.	
96.		212.	
97.	PROFESSION	213.	PROFESSION
98.	SUBINTEREST	214.	SUBINTEREST

99.		215.	
100.		216.	
101.	PROFESSION	217.	PROFESSION
102.	SUBINTEREST	218.	SUBINTEREST
103.		219.	
104.		220.	
105.	PROFESSION	221.	PROFESSION
106.	SUBINTEREST	222.	SUBINTEREST
107.		223.	
108.		224.	
109.	PROFESSION	225.	PROFESSION
110.	SUBINTEREST	226.	SUBINTEREST
111.		227.	
112.		228.	
113.	PROFESSION	229.	PROFESSION
114.	SUBINTEREST	230.	SUBINTEREST
115.		231.	
116.		232.	

VARIABLE LIST

1	Profession	NI
2	SubInterest	NI
3	Focus	NI
4	Area	NI

STEPS FOR IMPLEMENTING FORWARD CHAINING ERROR! BOOKMARK NOT DEFINED.

1. Variables are identified from the user's input using Keyword matching.
2. The variable is placed in the Conclusion-variable queue and the corresponding variable's value is instantiated in the Variable List.
3. Clause-variable List is searched for the variable present in front of the queue. If found, the corresponding rule no is found using the below formulae
   Rule = ((clauseNo/4) +1) *10
4. The calculated rule no and the clause no is placed in the clause-variable pointer.
5. Each variable in the IF part is instantiated (if not already) and the variable-list is updated accordingly.
6. If all the clauses are true in the IF part, then THEN part is invoked and the variable in the THEN part is added to the back of the conclusion variable queue.
7. When there are no more IF statements containing the variable present in front of the queue, then the front variable is removed from the conclusion variable list.
8. If there are no more variables in the conclusion variable queue, the session is ended and a report is presented to the user. If there are more variables then go to Step 3.

ANALYSIS

ANALYSIS OF THE SHARED C PROGRAM

- Inference engine logic and knowledge base are included in the same program which will lead to frequent changes for whole program whenever knowledge base is to be changed.
- The program is inefficient as "GOTO" statements are used in the program.
- Arrays are used store the list which is memory inefficient and not easy accessible.
- The object-oriented concepts can't be implemented as the program is written in C.
- Usage of global variables makes the variable less secured as there is no access control.
- Exceptions are not being handled in the program provided.
- Reusability of code is not possible as the data structures are hardcoded, cannot use variable list and conclusion list for both methodologies, which cannot be optimized in terms of memory
- The forward chaining process and backward chaining process has to be implemented separately.

MODIFICATION DONE IN THE IMPLEMENTED PROGRAM

- Opted Java Programming Language for better usage of generic data structures and implementing OOPS concepts. Constructed the code from scratch by understanding the logic provided from sample program
- Implementation of two classes, one for Forward-Chaining and the other for Backward-Chaining and have created a 3rd class that creates objects both the FC and BC and access the methods.
- The Knowledge Base and the Data-structures used by the FC and BC process are not hardcoded, we have only 1 input file each for BC and FC for knowledge base. The input file is accessed using files input/output. Any changes in the knowledge base will not impact whole program.
- Dynamic memory management has been done by using LinkedHashMap, Stack and List making the program efficient.
- Exception are handled separately using try/catch blocks for better error modification.
- Usage of Lambda Expressions for reduction of lines of code and set a block of code in a simple and efficient way.

PLATFORM FOR RUNNING THE PROGRAM IMPLEMENTED

- Use JavaDevelopmentKit8/JRE8 for execution of the program
- Use Netbeans8 IDE or Install it from the link provided here (https://netbeans.org/downloads/)
- Go to open project, select the project provided in ZIP file and execute it.

IMPLEMENTING INFERENCE PROGRAM IN JAVA

BackwardChaining.java

```
import java.util.*;

class BackwardChaining
{
    public List<String> KnBaseIFMap = new ArrayList<>();
    //If/else condition part of knowledge base

    public LinkedHashMap<Integer, String> KnBase = new LinkedHashMap<>();
    //Knowledge base

    public LinkedHashMap<Integer, String> ClauseVarList=new LinkedHashMap<>();
    //Clause Variable list

    public List<String> ConclVarList=new ArrayList<>();
    //Conclusion Variable list

    public LinkedHashMap<String, String> VarList=new LinkedHashMap<>();
    //Variable list

    public Stack<String> ConclusionStack = new Stack<>();
    //Conclusion Stack, holds the latest conclusion

    String profession= null;
```

```
public void generateDisplayBCKnowledgeBase()
   {
      //Loading up the if/else condition part of knowledge base
       KnBaseIFMap.add("Interest=Yes AND Technology=Yes THEN Profession=Engineer");
       KnBaseIFMap.add("Science=Yes AND HealthCare=Yes THEN Profession=HealthCare");
       KnBaseIFMap.add("Science=Yes AND HealthCare=No AND HumanAnotomy=Yes THEN
Profession=Doctor");
       KnBaseIFMap.add("Science=Yes AND HealthCare=No AND HumanAnotomy=No AND
Plants=Yes THEN Profession=Botanist");
       KnBaseIFMap.add("Science=Yes AND HealthCare=No AND HumanAnotomy=No AND
Plants=No AND Animals=Yes THEN Profession=Zoologist");
       KnBaseIFMap.add("Maths=Yes AND Shares=Yes  THEN Profession=Business");
       KnBaseIFMap.add("Maths=Yes AND Shares=No AND Logical=Yes  THEN
Profession=Scientist");
       KnBaseIFMap.add("Maths=Yes AND Shares=No AND Logical=No AND Management=Yes
THEN Profession=Manager");
       KnBaseIFMap.add("History=Yes AND culture_history=Yes  THEN Profession=Anthropologist");
       KnBaseIFMap.add("History=Yes AND culture_history=No AND Fossil=Yes  THEN
Profession=Palentologist");
       KnBaseIFMap.add(" Arts=Yes AND Performances=Yes  THEN Profession=Fine_Artist");
       KnBaseIFMap.add("Arts=Yes AND Performances=No AND Language=Yes  THEN
Profession=Communication_Specialist");


       //Loading up the knowledge base in format of "10 IF a=b THEN b=c; 20 IF a=c THEN c=d etc.,"
       int KnBaseindex=10;
       for (String entry : KnBaseIFMap)
       {
          KnBase.put(KnBaseindex, entry);
          KnBaseindex=KnBaseindex+10;
       }



   }
```

```
public void generateClauseVariableList()
{
    //Loading up the Clause Variable list in format of "1 var1; 2; 3 var2l 4; 5 var3 etc.," leaving six
places  b/w each rule
    int index=1;
    for(String entry : KnBase.values())
    {
        if(entry.split("AND").length > 0)
        {
            int HashIndex=0;
            for(int i=0;i<entry.split("AND").length;i++)
            {
                ClauseVarList.put(index, entry.split("AND")[i].split("=")[0].trim());
                index++;
                HashIndex++;
            }
            index=index+(6-HashIndex);
        }
    }

    //Loading up the conclusion varibale list <Rule No> <Conclusion> ex: 30 Profession from
Knowledge base
    for(Integer entry : KnBase.keySet())
    {
        ConclVarList.add(entry+"="+KnBase.get(entry).split("THEN")[1].trim().split("=")[0].trim());
    }

    List<String> tempList=ConclVarList;
    Collections.reverse(tempList); //Since stack is FIFO, reversing a temp list and feeding the stack

    for(String tempstr : tempList)
```

```
    {
        if(tempstr.split("=")[1].trim().equals("Profession"))
        {
            ConclusionStack.add(tempstr.split("=")[0].trim());
        }
    }

}

public void printVariableList()
{
    System.out.println("----------------");
    System.out.println("Knowledge Base");
    System.out.println("----------------");
    KnBase.forEach((i,j) -> System.out.println(i+" "+j));

    System.out.println("--------------------");
    System.out.println("Clause variable List");
    System.out.println("--------------------");
    ClauseVarList.forEach((i,j) -> System.out.println(i+" "+j));

    System.out.println("--------------------");
    System.out.println("Conclusion variable List");
    System.out.println("--------------------");
    ConclVarList.forEach((i) -> System.out.println(i));

    System.out.println("--------------------");
    System.out.println("Conclusion variable Stack");
    System.out.println("--------------------");
    ConclusionStack.forEach((i) -> System.out.println(i));
```

```
    }

    public String BC_Functionality()
    {
        Scanner input=new Scanner(System.in);
        VarList.put("Interest","Yes"); //Instantiating for the first time

        // 1. Picks the top element from the stack, finds out the clause variable by applying hash to Rule number
        // 2. iterates over all the clause variables and instantiate the variable table by asking user if already not present
        // 3. once every clause variable instantiated, executes the IF/THEN to provide the final answer
        while(ConclusionStack.empty() != true)
        {
            String Rule=ConclusionStack.peek();
            int Clause=(((Integer.parseInt(Rule)/10)-1)*6)+1; //hash to find the clause from Rule
            int IFClauseNo=0;

            //Setting up the VarList table
            for(int i=0; i<6; i++)
            {

                if(ClauseVarList.get(Clause) != null ) //checking if caluse var is not null
                {
                    IFClauseNo++;
                    if(VarList.get(ClauseVarList.get(Clause)) == null ) //ask user if the var list is not instatiated for this clause
                    {
                        System.out.println("Are you interested in "+ClauseVarList.get(Clause)+ "?[Yes/No]: ");
                        String tempChoice=input.next(); //user input for choice of interest
                        //instantiate the var list for this clause
```

```
                VarList.put(ClauseVarList.get(Clause), tempChoice);
            }

            if(i==0 && VarList.get(ClauseVarList.get(Clause)).equalsIgnoreCase("No"))
            {
                break;
            }
        }
        Clause++;
    }

    int AllIF=0;
    for(int i=0; i<IFClauseNo;i++) //execute if/then part for this rule
    {
if(VarList.get(KnBase.get(Integer.parseInt(Rule)).split("THEN")[0].trim().split("AND")[i].trim().split("=")[0]).equalsIgnoreCase(KnBase.get(Integer.parseInt(Rule)).split("THEN")[0].trim().split("AND")[i].trim().split("=")[1]))
        {
          AllIF++;
        }
    }

    if(AllIF == IFClauseNo) //if all clause variables satisfies the given condition
    {
        profession = KnBase.get(Integer.parseInt(Rule)).split("THEN")[1].trim().split("=")[1];
        System.out.println("Ur prof is: "+ profession); //hurray...!
        break;
    }
    else
    {
```

```
            ConclusionStack.pop(); //this rule is not valid for the current choice of interest, hence popping.

        }

    }
    if(profession==null)
    {
        System.out.println("Sorry !!! We dont have any other Interest Constructed just yet !!!");
        System.out.println("You could re-execute and start over !!!");
    }
    return profession;
  }
}
```

ForwardChaining.java

```
import java.util.*;
import java.nio.file.Files;
import java.nio.file.Paths;

public class ForwardChaining {
  //Creating an LinkedHashMap Object for the Storage of If/else condition part of knowledge base
  LinkedHashMap<String, String> KnBaseIFMap = new LinkedHashMap<>();

  //Creating another LinkedHashMap Object to store the rule no and the IF-THEN Rules.
  LinkedHashMap<Integer, LinkedHashMap<String, String>> KnBase = new LinkedHashMap<>();

  //Creating another LinkedHashMap Object to store Clause Variable list
  LinkedHashMap<Integer, String> ClauseVarList=new LinkedHashMap<>();
```

```
//Conlusion variable queue
Queue<String> ConclVarListQ = new LinkedList<>();

//Variable list
LinkedHashMap<String, String> VarList=new LinkedHashMap<>();

//A list created to store the subInterest for further processing
List<String> l= new ArrayList<>();

public void generate_display_KnowledgeBase()
{
   //The Knowledge Base is generated from the input file and stored in the LinkedHashMap object
   try{
   Files.lines(Paths.get("input.txt")).forEach(
   KB -> {
       String[] KBSplit = KB.split(" THEN ");
       KnBaseIFMap.put(KBSplit[0], KBSplit[1]);
       l.add(KBSplit[0].split("SubInterest=")[1].split(" ")[0]);
      });
   }catch(Exception e){
   System.out.println("Caught Exception in generate_display_KnowledgeBase function: " + e);
   }

    //Loading up the knowledge base in format of "10 IF a=b THEN b=c; 20 IF a=c THEN c=d etc.,"
    int KnBaseindex=10;
    for (Map.Entry<String, String> entry : KnBaseIFMap.entrySet())
    {
        //Displaying the rules in the form of IF THEN for the convience of this project.
        //System.out.println(KnBaseindex + " IF " + entry.getKey() + " THEN " + entry.getValue());
```

```
            LinkedHashMap temp=new LinkedHashMap<>();
            temp.put(entry.getKey(), entry.getValue());
            //This object stores the calculated rule no and the corresponding object that contains IF THEN
rules
            KnBase.put(KnBaseindex, temp);
            KnBaseindex=KnBaseindex+10;
        }
    }


    public void generate_ClauseVariableList()
    {
        //Loading up the Clause Variable list in format of "1 var1; 2; 3 var2l 4; 5 var3 etc.," leaving two
places to b/w each variable
        int index=1;
        for(Map<String, String> entry : KnBase.values())//Outer loop to take value part of Hash object
knBaseIFMAP
        {
            for(Map.Entry<String, String> MapEntry : entry.entrySet())//Inner loop to take the Key part of the
hash object knBaseIFMap
            {
                String[] s = MapEntry.getKey().toString().split("&");//Splits the String based on &
                int k=0;
                for(String s1:s)
                {
                    ClauseVarList.put(index, s1.split("=")[0]); //Re-splits the above string again based on =
                    index=index+1;
                    k++;
                }
                if(k==2) index=index+2; //If there are 2 clauses in IF part, this is executed.
                if(k==3) index=index+1; //If there are 3 clauses in If part, this is executed.
            }
        }
```

```
        //Displays the ClauseVariable List for FC
        //display_ClauseVariableList();
    }

    public void FC_functionality(String initialInput)
    {
        String subInterest, subIn="No";
        System.out.println();
        System.out.println("--------------------Your FC Process Starts---------------------------");
        System.out.println();
        System.out.println("From BC Process - Your advised Profession is : " + initialInput);

        ConclVarListQ.add("Profession"); //Raw input from user/output of the Backwards Chaining
inference
        //displayConclusionVariableQueue();

        VarList.put("Profession",initialInput); //Instantiating for the first time
        //displayVariableList();



        while(ConclVarListQ.isEmpty() != true)
        {
            String tempstr=ConclVarListQ.peek();//Takes the first value of the ConclusionVariable queue
            int RuleNo;
           for(Map.Entry<Integer, String> entry : ClauseVarList.entrySet())
            {
                if(entry.getValue().trim().equals(tempstr.trim()))
                {
                    RuleNo=((entry.getKey()/4)+1)*10; //the logic to get the RuleNo
                    //System.out.println("Checking Rule No : " + RuleNo);
```

```
//Below lines has the logic that iterates over clause variable list and get the corresponding
conclusion variable values from the Knowledge base and instantiate them in Variable list)
LinkedHashMap<String, String> tempKnBaseIFMap = KnBase.get(RuleNo);
for(Map.Entry<String, String> tempKBMap : tempKnBaseIFMap.entrySet())
{
   String[] s = tempKBMap.getKey().split("&");
   String[] sub = s[0].split("=");

   if(sub[0].trim().equals(tempstr) && sub[1].trim().equals(VarList.get(tempstr))) //If
variable in rule matches the one present on top of the queue
   {
      if(VarList.containsKey(s[1].split("=")[0]))
         {//Do Nothing
         }
      else if(VarList.containsKey(s[1].split("=")[0]) == false)
      {
         Scanner in = new Scanner(System.in);
         List<String> listTemp = new ArrayList<>();;
         //Implemented different cases to handle different sub-Profession
         switch(VarList.get(sub[0].trim()))
         {
            //Takes only a part of the list that belong to that Profession

            case "Engineer":    listTemp = l.subList(0, 5);
                                break;
            case "HealthCare" : listTemp = l.subList(5, 10);
                                break;
            case "Doctor" :     listTemp = l.subList(10, 15);
                                break;
            case "Botanist" :   listTemp = l.subList(16, 19);
```

```
                    break;
            case "Zoologist" :  listTemp = l.subList(21, 26);
                    break;
            case "Management" : listTemp = l.subList(26, 32);
                    break;
            case "Business" :   listTemp = l.subList(32, 38);
                    break;
            case "Palentologist" :  listTemp = l.subList(38, 43);
                        break;
            case "Anthropologist" : listTemp = l.subList(43, 48);
                        break;
            case "Fine_Artist" :    listTemp = l.subList(48, 53);
                        break;
            case "Communication_Specialist":    listTemp = l.subList(53, 58);
                                break;
            case "Scientist":   listTemp = l.subList(58,63);
                                break;
            default : System.out.println("Sorry !! We dont have an specialised area constructed
on your advised Profession just yet.");
        }
        try
        {
            //Based on the above sub-list, it iterates over the list to ask the user back for
appropriate query
            Iterator itr = listTemp.iterator();
            do
            {
                subInterest=(String)itr.next();
                System.out.println("Do you like " + subInterest + "? [Yes/No] :");
                subIn = in.next();
            }while(subIn.equalsIgnoreCase("No") && itr.hasNext());
```

```
                    if(subIn.equalsIgnoreCase("Yes"))
                    {
                        //If user enters yes, then they are added to Variable List and Conclusion
Variable Queue.
                        VarList.put(s[1].split("=")[0], subInterest);
                        ConclVarListQ.add(s[1].split("=")[0]);
                        //Below are extra code of lines for certain rules that has more than 2 clauses
                        if(VarList.get(sub[0].trim()).equalsIgnoreCase("Botanist"))
                        {
                            if(subInterest.equalsIgnoreCase("Grow_Plants"))
                            {
                                System.out.println("Do you like Growing Small or Large Scale?
[Small/Large]: ");
                                String tempInput=in.next();
                                if(tempInput.equalsIgnoreCase("Small"))
                                {
                                    VarList.put(s[2].split("=")[0],"Small");
                                    ConclVarListQ.add(s[2].split("=")[0]);
                                }
                                else if(tempInput.equalsIgnoreCase("Large"))
                                {
                                    VarList.put(s[2].split("=")[0],"Large");
                                    ConclVarListQ.add(s[2].split("=")[0]);
                                }
                            }
                            if(subInterest.equalsIgnoreCase("Study_Plants"))
                            {
                                System.out.println("Do you like Studying Internal or External Strcuture?
[Internal/External]: ");
                                String tempInput=in.next();
                                if(tempInput.equalsIgnoreCase("External"))
```

```
                    {
                        VarList.put(s[2].split("=")[0],"External_Study");
                        ConclVarListQ.add(s[2].split("=")[0]);
                    }
                    else if(tempInput.equalsIgnoreCase("Internal"))
                    {
                        VarList.put(s[2].split("=")[0],"Internal_Study");
                        ConclVarListQ.add(s[2].split("=")[0]);
                    }
                }
            }
        }
        else
            System.out.println("I am sorry. We dont have an specialised area constructed
on your Interests just yet.");

        }catch(Exception e){
            System.out.println("Caught Exception Inside : " +e);
        }
    }
  }
 }

 //displayConclusionVariableQueue();
 //displayVariableList();
}
}
//Removing the element from front queue once the iteration of knwoledge base is compelete
ConclVarListQ.remove();
//System.out.println("After Removing from Queue");
```

```
        //displayConclusionVariableQueue();
    }
}

public void displayConclusionVariableQueue()
{
    //This function displays the Concusion Variable Queue to the user
    System.out.println("------------------------------");
    System.out.println("----FC Conclusion Var queue----");
    System.out.println("------------------------------");
    ConclVarListQ.forEach(i -> System.out.println(i));
    System.out.println();
}

public void displayVariableList()
{
    //This function displays the Varliable List to the user
    System.out.println("------------------------------");
    System.out.println("--------FC Variable List------");
    System.out.println("------------------------------");
    VarList.forEach((i,j) -> System.out.println(i+"="+j));
    System.out.println();
}

public void display_ClauseVariableList()
{
    //This function displays the Clause-Variable List to th user
    System.out.println("---------------------------------");
    System.out.println("-----FC Clause variable List-----");
    System.out.println("---------------------------------");
```

```
        ClauseVarList.forEach((i,j) -> System.out.println(i+" "+j)); //Prints the ClauseVariable List obtained above
        System.out.println();
    }

    public void display_KnowledgeBase()
    {
        //This displays Knowledge base to the user
        System.out.println("------------------------------");
        System.out.println("-------FC Knowledge Base------");
        System.out.println("------------------------------");
        KnBase.forEach((i,j) -> System.out.println(i+" "+j));
    }

    public void finalInput()
    {
        //This function displays the final output to the user. It compares the Variables in the Variable List object with the rules stores in KnBaseIFMap object and finds the appropriate conclusion
        System.out.println("======End of the Inference=======");
        System.out.println();
        System.out.println();
        System.out.println("FC inference says : ");
        String[] value = new String[3];
        int k=0;
        //Prints the value from the VarList first
        for(Map.Entry<String, String> entry : VarList.entrySet()) {
            System.out.println(entry.getKey() + " is " + entry.getValue());
            value[k]=entry.getValue().trim();
            k++;
        };
        final int c=k;
```

```
        //Compares the values in VarList with the Knowledge Storage to give final output.
        KnBaseIFMap.forEach((m,n) ->
            {
                String[] s = m.split("&");
                String[] t = n.split("=");
                if(c == 2)
                {
                    if(s[0].split("=")[1].trim().equals(value[0]) && s[1].split("=")[1].trim().equals(value[1]))
                    {
                        System.out.println("Hence Your Advised Specialised Area is: " + t[1]);
                        System.out.println();
                    }
                }
                else if(c == 3)
                {
                    if(s[0].split("=")[1].trim().equals(value[0]) && s[1].split("=")[1].trim().equals(value[1]))
                    {
                        if((s[2].split("=")[1].trim().equals(value[2])))
                        {
                            System.out.println("Hence Your Advised Specialised Area is: " + t[1]);
                            System.out.println();
                        }
                    }
                }
            });
    }
}
```

BC_FC_Demo.java

```
import java.util.*;

class BC_FC_Demo{
    public static void main(String args[])
    {
        System.out.print("\033[H\033[2J");
        System.out.flush();

        BackwardChaining bc = new BackwardChaining();
        ForwardChaining fc = new ForwardChaining();
        //This function will call the BC functionality
        System.out.println("-----------------------------------------------------");
        System.out.println("Welcome to Career Advising Support Service");
        System.out.println("-----------------------------------------------------");
        System.out.println();
        System.out.println("This service is very easy to use.....");
        System.out.println("You need to answer to a couple of questions and we will advise you based on
your answers");
        System.out.println();
        Scanner input =new Scanner(System.in);
        try
        {

            System.out.println("Are you ready ? [Yes/No] : ");
            String str = input.nextLine();
            System.out.println();
```

```
        if(str.equalsIgnoreCase("Yes"))
        {
            //This function generate Knowledge base
            bc.generateDisplayBCKnowledgeBase();
            //This functional Generates the Clause-variable List from Knowledge base and displays
variable, conclusion variable and clause variable list
            bc.generateClauseVariableList();
            //Displays the final output for the user based on its inferences.
            String profession = bc.BC_Functionality();
            //This function generate Knowledge base from the Input File and Displays the same to the user
            fc.generate_display_KnowledgeBase();
            //This functional Generates the Clause-variable List from Knowledge base and displays the
same
            fc.generate_ClauseVariableList();
            //This functionality involves the actual FC inference Engine's work.
            if(profession!=null)
              {
                fc.FC_functionality(profession);
                //Displays the final output for the user based on its inferences.
                fc.finalInput();
                //Display the KnowledgeBase, ClauseVariable and Variable List if needed
                System.out.println("Before Quitting Do you wish to see the Knowledge base, clause
variable list and Variable List generated/used during BC and FC [Yes/No]:");
                String ans = input.next();
                System.out.println();
                if(ans.equalsIgnoreCase("Yes"))
                {
                  bc.printVariableList();

System.out.println("*********************************************************************
******");
                  fc.display_KnowledgeBase();
```

```
                    fc.display_ClauseVariableList();
                    fc.displayVariableList();
                }
            }
        System.out.println();
        System.out.println("THANK YOU !!");
        System.out.println();
    }
    else
        System.out.println("Looks like you are not ready. U can re-execute when ready !! Thank You !!!");

    }catch(Exception e){
        System.out.println("Caught Exception in Main" + e);
    }
  }

}
```

RESULT

OUTPUT1:

IF Technology=Yes THEN Profession=Engineer"

IF Profession = Engineer and SubInterest=Construction THEN Area=Civil_Engineering

IF Profession = Engineer and SubInterest=Computers THEN Area=Computer_Engineering

IF Profession = Engineer and SubInterest=Chemicals THEN Area=Chemical_Engineering

IF Profession = Engineer and SubInterest=Machines THEN Area=Mechanical_Engineering

IF Profession = Engineer and SubInterest=Electronics THEN Area=Electrical_Engineering

```
Output
Debugger Console   BC_FC_Demo (run)
run:
-------------------------------------------------------
Welcome to Career Advising Support Service
-------------------------------------------------------

This service is very easy to use.....
You need to answer to a couple of questions and we will advise you based on your answers

Are you ready ? [Yes/No] :
yes

Are you interested in Technology?[Yes/No]:
Yes
Ur prof is: Engineer

--------------------Your FC Process Starts---------------------------

From BC Process - Your advised Profession is : Engineer
Do you like Construction? [Yes/No] :
no
Do you like Computers? [Yes/No] :
yes
======End of the Inference=======


FC inference says :
Profession is Engineer
 SubInterest is Computers
Hence Your Advised Specialised Area is: Computer_Engineering

Before Quitting Do you wish to see the Knowledge base, clause variable list and Variable List generated/used during BC and FC [Yes/No]:
no


THANK YOU !!

BUILD SUCCESSFUL (total time: 25 seconds)
```

OUTPUT2:

IF Maths=Yes AND Shares=Yes THEN Profession=Business"

IF Profession = Business & SubInterest=Finance THEN Area=Accountant

IF Profession = Business & SubInterest=Sales THEN Area=Marketing

IF Profession = Business & SubInterest=Manufacturing THEN Area=Manufactuing_Business

IF Profession = Business & SubInterest=Merchandise THEN Area=Merchandising_Business

IF Profession = Business & SubInterest=Business_Law THEN Area=Business_Lawyer

IF Profession = Business & SubInterest=Cross_Border_Trade THEN Area=International_Business

```
Output
Debugger Console   BC_FC_Demo (run)
This service is very easy to use.....
You need to answer to a couple of questions and we will advise you based on your answers

Are you ready ? [Yes/No] :
yes

Are you interested in Technology?[Yes/No]:
no
Are you interested in Science?[Yes/No]:
no
Are you interested in Maths?[Yes/No]:
yes
Are you interested in Shares?[Yes/No]:
yes
Ur prof is: Business

--------------------Your FC Process Starts--------------------------

From BC Process - Your advised Profession is : Business
Do you like Finance? [Yes/No] :
no
Do you like Sales? [Yes/No] :
no
Do you like Manufacturing? [Yes/No] :
no
Do you like Merchandise? [Yes/No] :
yes
======End of the Inference=======


FC inference says :
Profession is Business
 SubInterest is Merchandise
Hence Your Advised Specialised Area is: Merchandising_Business

Before Quitting Do you wish to see the Knowledge base, clause variable list and Variable List generated/used during BC and FC [Yes/No]:
no


THANK YOU !!

BUILD SUCCESSFUL (total time: 28 seconds)
```

OUTPUT3

If Science=Yes AND Healthcare=No AND HumanAnotomy=No AND Plants=No AND Animals=Yes THEN Profession=Zoologist

IF Profession = Zoologist & SubInterest=Animal_Health THEN Area=Animal_Veteran

IF Profession = Zoologist & SubInterest=Study_Mammals THEN Area=Mammology

IF Profession = Zoologist & SubInterest=Study_Birds THEN Area=Ornitholgy

IF Profession = Zoologist & SubInterest=Marine_Animals THEN Area=Marine_Biologist

IF Profession = Zoologist & SubInterest=Study_Reptiles THEN Area=Herpetology

```
Output
Debugger Console   BC_FC_Demo (run)
run:
-------------------------------------------------------
Welcome to Career Advising Support Service
-------------------------------------------------------

This service is very easy to use.....
You need to answer to a couple of questions and we will advise you based on your answers

Are you ready ? [Yes/No] :
yes

Are you interested in Technology?[Yes/No]:
no
Are you interested in Science?[Yes/No]:
yes
Are you interested in HealthCare?[Yes/No]:
no
Are you interested in HumanAnotomy?[Yes/No]:
no
Are you interested in Plants?[Yes/No]:
no
Are you interested in Animals?[Yes/No]:
yes
Ur prof is: Zoologist

--------------------Your FC Process Starts---------------------------

From BC Process - Your advised Profession is : Zoologist
Do you like Animal_Health? [Yes/No] :
no
Do you like Study_Mammals? [Yes/No] :
no
Do you like Study_Birds? [Yes/No] :
yes
======End of the Inference=======


FC inference says :
Profession is Zoologist
 SubInterest is Study_Birds
Hence Your Advised Specialised Area is: Ornitholgy

Before Quitting Do you wish to see the Knowledge base, clause variable list and Variable List generated/used during BC and FC [Yes/No]:
no


THANK YOU !!
```

OUTPUT4

If HISTORY = YES And CULTURE_HISTORY = NO And FOSSILS = YES Then PROFESSION =Paleontologist.

IF Profession = Paleontologist and SubInterest=Fossilized_Plants THEN Area=Paleo_Botany

IF Profession = Paleontologist and SubInterest=Fossilized_Animals and Focus=Invertebrate THEN Area=Invertebrate_Paleontology

IF Profession = Paleontologist and SubInterest=Fossilized_Animals and Focus=Vertebrate THEN Area=Vertebrate_Paleontology

IF Profession = Paleontologist and SubInterest=Preserve_Fossils THEN Area=Taphonomy

```
run:
-----------------------------------------------------
Welcome to Career Advising Support Service
-----------------------------------------------------

This service is very easy to use.....
You need to answer to a couple of questions and we will advise you based on your answers

Are you ready ? [Yes/No] :
yes

Are you interested in Technology?[Yes/No]:
no
Are you interested in Science?[Yes/No]:
no
Are you interested in Maths?[Yes/No]:
no
Are you interested in History?[Yes/No]:
yes
Are you interested in culture_history?[Yes/No]:
no
Are you interested in Fossil?[Yes/No]:
yes
Ur prof is: Palentologist

--------------------Your FC Process Starts--------------------------

From BC Process - Your advised Profession is : Palentologist
Do you like Mircoscopic_Fossils? [Yes/No] :
no
Do you like Fossilized_Plants? [Yes/No] :
yes
======End of the Inference=======


FC inference says :
Profession is Palentologist
 SubInterest is Fossilized_Plants
Hence Your Advised Specialised Area is: Paleo_Botany

Before Quitting Do you wish to see the Knowledge base, clause variable list and Variable List generated/used during BC and FC [Yes/No]:
no


THANK YOU !!
```

OUTPUT5(DISPLAY KNOWLEDGE BASE AND CLAUSE VARIABLE LIST)

If ARTS = YES And MUSIC = YES Then PROFESSION =Fine_Artist.

IF Profession = Fine_Artist & SubInterest=Music THEN Area=Musician

IF Profession = Fine_Artist & SubInterest= Carving THEN Area=Sculpturer

IF Profession = Fine_Artist & SubInterest=Dance THEN Area=Dance

IF Profession = Fine_Artist & SubInterest=Cinematography THEN Area=Film_Media

IF Profession = Fine_Artist & SubInterest=Capture_Moments THEN Area=Photographer

```
Output - BC_FC_Demo (run)
run:
-------------------------------------------------------
Welcome to Career Advising Support Service
-------------------------------------------------------

This service is very easy to use.....
You need to answer to a couple of questions and we will advise you based on your answers

Are you ready ? [Yes/No] :
yes

Are you interested in Technology?[Yes/No]:
no
Are you interested in Science?[Yes/No]:
no
Are you interested in Maths?[Yes/No]:
no
Are you interested in History?[Yes/No]:
no
Are you interested in Arts?[Yes/No]:
yes
Are you interested in Performances?[Yes/No]:
yes
Ur prof is: Fine_Artist

--------------------Your FC Process Starts---------------------------

From BC Process - Your advised Profession is : Fine_Artist
Do you like Music? [Yes/No] :
no
Do you like Carving? [Yes/No] :
no
Do you like Dance? [Yes/No] :
no
Do you like Cinematography? [Yes/No] :
yes
======End of the Inference=======


FC inference says :
Profession is Fine_Artist
 SubInterest is Cinematography
Hence Your Advised Specialised Area is: Film_Media
```

```
Before Quitting Do you wish to see the Knowledge base, clause variable list and Variable List generated/used during BC and FC [Yes/No]:
yes

----------------
Knowledge Base
----------------
10 Interest=Yes AND Technology=Yes THEN Profession=Engineer
20 Science=Yes AND HealthCare=Yes THEN Profession=HealthCare
30 Science=Yes AND HealthCare=No AND HumanAnotomy=Yes THEN Profession=Doctor
40 Science=Yes AND HealthCare=No AND HumanAnotomy=No AND Plants=Yes THEN Profession=Botanist
50 Science=Yes AND HealthCare=No AND HumanAnotomy=No AND Plants=No AND Animals=Yes THEN Profession=Zoologist
60 Maths=Yes AND Shares=Yes  THEN Profession=Business
70 Maths=Yes AND Shares=No AND Logical=Yes  THEN Profession=Scientist
80 Maths=Yes AND Shares=No AND Logical=No AND Management=Yes  THEN Profession=Manager
90 History=Yes AND culture_history=Yes  THEN Profession=Anthropologist
100 History=Yes AND culture_history=No AND Fossil=Yes  THEN Profession=Palentologist
110  Arts=Yes AND Performances=Yes  THEN Profession=Fine_Artist
120 Arts=Yes AND Performances=No AND Language=Yes  THEN Profession=Communication_Specialist
--------------------
Clause variable List
--------------------
1 Interest
2 Technology
7 Science
8 HealthCare
13 Science
14 HealthCare
15 HumanAnotomy
19 Science
20 HealthCare
21 HumanAnotomy
22 Plants
25 Science
26 HealthCare
27 HumanAnotomy
28 Plants
29 Animals
31 Maths
32 Shares
37 Maths
38 Shares
39 Logical
43 Maths
44 Shares
```

Output - BC_FC_Demo (run)

```
43 Maths
44 Shares
45 Logical
46 Management
49 History
50 culture_history
55 History
56 culture_history
57 Fossil
61 Arts
62 Performances
67 Arts
68 Performances
69 Language
--------------------
Conclusion variable List
--------------------
120=Profession
110=Profession
100=Profession
90=Profession
80=Profession
70=Profession
60=Profession
50=Profession
40=Profession
30=Profession
20=Profession
10=Profession
--------------------
Conclusion variable Stack
--------------------
120
110
*****************************************************************************
------------------------------
-------FC Knowledge Base------
------------------------------
10 {Profession = Engineer & SubInterest=Construction=Area=Civil_Engineering}
20 {Profession = Engineer & SubInterest=Computers=Area=Computer_Engineering}
30 {Profession = Engineer & SubInterest=Chemicals=Area=Chemical_Engineering}
40 {Profession = Engineer & SubInterest=Machines=Area=Mechanical_Engineering}
50 {Profession = Engineer & SubInterest=Electronics=Area=Electrical_Engineering}
60 {Profession = HealthCare & SubInterest=Pharmacy=Area=Pharmacist}
```

Output - BC_FC_Demo (run)

```
50 {Profession = Engineer & SubInterest=Electronics=Area=Electrical_Engineering}
60 {Profession = HealthCare & SubInterest=Pharmacy=Area=Pharmacist}
70 {Profession = HealthCare & SubInterest=Phycology=Area=Psychiatrist}
80 {Profession = HealthCare & SubInterest=Nursing=Area=Nurse}
90 {Profession = HealthCare & SubInterest=Imaging=Area=Radiologic_Technologist}
100 {Profession = HealthCare & SubInterest=Physiotherapy=Area=Physiotherapist}
110 {Profession = Doctor & SubInterest=Heart=Area=Cardiologist}
120 {Profession = Doctor & SubInterest=Teeth=Area=Dentist}
130 {Profession = Doctor & SubInterest=Bones=Area=Orthopedics}
140 {Profession = Doctor & SubInterest=Nerves=Area=Neurologist}
150 {Profession = Doctor & SubInterest=Children_General_Health=Area=Pediatrician}
160 {Profession = Botanist & SubInterest=Grow_Plants & Focus=Small=Area=Horticulture}
170 {Profession = Botanist & SubInterest=Grow_Plants & Focus=Large=Area=Agriculture}
180 {Profession = Botanist & SubInterest=Plant_Hybrids=Area=Agricultural_Hybridization}
190 {Profession = Botanist & SubInterest=Study_Plants & Focus=Internal_Study=Area=Plant_Anatomy}
200 {Profession = Botanist & SubInterest=Study_Plants & Focus=External_Study=Area=Plant_Morphlogy}
210 {Profession = Botanist & SubInterest=Diseased_Plant=Area=Plant_Pathology}
220 {Profession = Zoologist & SubInterest=Animal_Health=Area=Animal_Veteran}
230 {Profession = Zoologist & SubInterest=Study_Mammals=Area=Mammology}
240 {Profession = Zoologist & SubInterest=Study_Birds=Area=Ornitholgy}
250 {Profession = Zoologist & SubInterest=Marine_Animals=Area=Marine_Biologist}
260 {Profession = Zoologist & SubInterest=Study_Reptiles=Area=Herpetology}
270 {Profession = Management & SubInterest=Oversee_Hospitality=Area=Hospitality_Management}
280 {Profession = Management & SubInterest=Oversee_Hotel=Area=Hotel_Management}
290 {Profession = Management & SubInterest=Oversee_New_Product_Development=Area=R&D_Management}
300 {Profession = Management & SubInterest=Oversee_Staffing=Area=HR_Management}
310 {Profession = Management & SubInterest=Oversee_Long_Distance_Expedition=Area=Travel&Tour_Management
320 {Profession = Management & SubInterest=Oversee_Budget=Area=Budget_Management}
330 {Profession = Business & SubInterest=Finance=Area=Accountant}
340 {Profession = Business & SubInterest=Sales=Area=Marketing}
350 {Profession = Business & SubInterest=Manufacturing=Area=Manufactuing_Business}
360 {Profession = Business & SubInterest=Merchandise=Area=Merchandising_Business}
370 {Profession = Business & SubInterest=Business_Law=Area=Business_Lawyer}
380 {Profession = Business & SubInterest=Cross_Border_Trade=Area=International_Business}
390 {Profession = Palentologist & SubInterest=Mircoscopic_Fossils=Area=Micropaleontology}
400 {Profession = Palentologist & SubInterest=Fossilized_Plants=Area=Paleo_Botany}
410 {Profession = Palentologist & SubInterest=Fossilized_Invertebrate_Animals=Area=Invertebrate_Paleont
420 {Profession = Palentologist & SubInterest=Fossilized_Vertebrate_Animals=Area=Vertebrate_Paleontolog
430 {Profession = Palentologist & SubInterest=Preserve_Fossils=Area=Taphonomy}
440 {Profession = Anthropologist & SubInterest=Pre-historic_Material_Culture_Study=Area=Archaeology}
450 {Profession = Anthropologist & SubInterest=Language_Origins=Area=Linguistic_Anthropology}
460 {Profession = Anthropologist & SubInterest=Human_Society_Study=Area=Cultural_Anthropology}
470 {Profession = Anthropologist & SubInterest=Study_Human_Evolution=Area=Paeo_Anthropology}
480 {Profession = Anthropologist & SubInterest=Study_Human_Remains_DNA=Area=Forensic_Anthropology}
490 {Profession = Fine Artist & SubInterest=Music=Area=Musician}
```

```
Output - BC_FC_Demo (run)
490 {Profession = Fine_Artist & SubInterest=Music=Area=Musician}
500 {Profession = Fine_Artist & SubInterest=Carving=Area=Sculpturer}
510 {Profession = Fine_Artist & SubInterest=Dance=Area=Dance}
520 {Profession = Fine_Artist & SubInterest=Cinematography=Area=Film_Media}
530 {Profession = Fine_Artist & SubInterest=Capture_Moments=Area=Photographer}
540 {Profession = Communication_Specialist & SubInterest=Build_Client_Reputation=Area=Public_Relations}
550 {Profession = Communication_Specialist & SubInterest=Promote_Product_Creatively=Area=Advertising}
560 {Profession = Communication_Specialist & SubInterest=Broadcast_News=Area=Journalisms}
570 {Profession = Communication_Specialist & SubInterest=Learn_Foreign_Lanaguages_Cultures=Area=International_Relations}
580 {Profession = Communication_Specialist & SubInterest=Teach_Interact_Students=Area=Professor}
590 {Profession = Scientist & SubInterest=Scientifc_Study_of_Universe=Area=Cosmologist}
600 {Profession = Scientist & SubInterest=Scientifc_Study_of_Nuclear_Process=Area=Nuclear_Physicist}
610 {Profession = Scientist & SubInterest=Scientifc_Study_of_Plasma=Area=Plasma_Physicist}
620 {Profession = Scientist & SubInterest=Scientifc_Study_of_Rocket_Science=Area=Rocket_Scientist}
630 {Profession = Scientist & SubInterest=Scientifc_Study_of_Quantum_Science=Area=Quantum_Physicist}
---------------------------------
-----FC Clause variable List-----
---------------------------------
1 Profession
2  SubInterest
5 Profession
6  SubInterest
9 Profession
10  SubInterest
13 Profession
14  SubInterest
17 Profession
18  SubInterest
21 Profession
22  SubInterest
25 Profession
26  SubInterest
29 Profession
30  SubInterest
33 Profession
34  SubInterest
37 Profession
38  SubInterest
41 Profession
42  SubInterest
```

Output - BC_FC_Demo (run)

```
197 Profession
198  SubInterest
201 Profession
202  SubInterest
205 Profession
206  SubInterest
209 Profession
210  SubInterest
213 Profession
214  SubInterest
217 Profession
218  SubInterest
221 Profession
222  SubInterest
225 Profession
226  SubInterest
229 Profession
230  SubInterest
233 Profession
234  SubInterest
237 Profession
238  SubInterest
241 Profession
242  SubInterest
245 Profession
246  SubInterest
249 Profession
250  SubInterest

------------------------------
--------FC Variable List------
------------------------------
Profession=Fine_Artist
 SubInterest=Cinematography


THANK YOU !!

BUILD SUCCESSFUL (total time: 48 seconds)
```

Analysis of the Results

- The main function initially reads the input text containing the knowledge of BC and generates the Knowledge base if-map for BC using Linked Hash Map.
- It then loads the Clause-Variable List for BC.
- It then loads the Conclusion List for BC
- The BC functionality is now invoked where-in, the top element of the inputted conclusion stack is looked for in the conclusion list. If found, the corresponding rule is removed from conclusion stack.
- The clause no is next computed using the formula provided in the Algorithm and get the values of those corresponding variables from the users which are not instantiated. This process continues until there are no more entries in conclusion stack and conclusion is initialized.
- Finally, the profession is advised to the user.
- Now the system prompts if the user wishes to know what specialized area under the advised profession.
- If the user enters yes, then the profession becomes the input to the FC process. In other words, the output of the BC process now becomes the input of the FC process.
- When the FC function is invoked, a set of actions mentioned in Step1 is executed except that this is for FC (so the knowledge base is different for FC).
- Similarly, the Clause-Variable list for FC is loaded.
- The input (in other words Profession in our case) is placed in conclusion variable queue.
- The first entry in the queue is searched and compared with the clause-variable list. If found, the corresponding rule no is calculated and the corresponding rule is executed.
- During the execution, if it encounters any variables that are not already instantiated, it requests the same from the users with appropriate prompts. Those entries are added to end of the conclusion variable queue for further process.
- This process continues until all there are no entries in the conclusion-variable queue.
- A final check is made to check if all the clauses in the rule are true, if true, then the corresponding THEN is invoked and the result is presented to the user.