

Subject- Paradigms and Computer Programming Fundamentals

Syllabus for IA1

Module 1-Introduction to Programming Paradigms and Core Language Design Issues

Module 2-Imperative Paradigm: Data Abstraction in Object Orientation

Module 4-Prolog

Practice Question For IA1

Module 1:Introduction to Programming Paradigms and Core Language Design Issues

- 1.Distinguish between imperative languages and declarative languages.
- 2.Explain static binding.
- 3..Explain dynamic binding.
- 4.Compare Compiler time binding & Run time binding(Early & Late Binding)
- 5.What is JVM.
- 6.Explain different types of programming paradigms.
- 7.What is dangling reference.
- 8.What is binding time.
- 9.What is scope.
- 10.Short note on Scope Rule,Static Scoping & dynamic Scoping .
- 11.Explain term Lifetime.
- 12.Explain following terms

- 1.Language design time
2. Language implementation time
3. program writing time
4. compile time
5. link time
6. load time
7. Runtime

- 13.Differentiate between compiler & interpreter

- 14. List different imperative languages.
- 15. List different declarative languages.

16. Explain following storage allocation with example

- 1. Static Allocation
- 2. Stack Allocation
- 3. Heap Allocation

17. What is being printed in following code assuming dynamic Scope & Static scope

```
int a=10;
fun1(int b)
{
    printf("%d", (a*b));
}
fun2()
{
    int a=20;
    fun1(30);
}
main()
{
    fun2();
}
Module 2
```

Module 2: Imperative Paradigm: Data Abstraction in Object Orientation

1) What are different features of object oriented programming paradigms?

- 2) What is inheritance? Explain different types of inheritance with example?
- 3) Explain polymorphism and its type.
- 4) How does C++ support the properties of encapsulation and data hiding?
- 5) What is virtual function and how it can be used in polymorphism?
- 6) What is constructor and its type in C++?

Module 4: Prolog

Exercise 1. How many facts, rules, clauses, and predicates are there in the following knowledge base? What are the heads of the rules, and what are the goals they contain?

```
woman(vincent).
woman(mia).
man(jules).
person(X):- man(X); woman(X).
loves(X,Y):- father(X,Y).
```

father(Y,Z):- man(Y), son(Z,Y).
father(Y,Z):- man(Y), daughter(Z,Y).

Exercise 2 Represent the following in Prolog:

1. Butch is a killer.
2. Mia and Marsellus are married.
3. Zed is dead.
4. Marsellus kills everyone who gives Mia a footmassage.
5. Mia loves everyone who is a good dancer.
6. Jules eats anything that is nutritious or tasty.

Exercise 3 Suppose we are working with the following knowledge base:

wizard(ron).
hasWand(harry).
quidditchPlayer(harry).
wizard(X):- hasBroom(X), hasWand(X).
hasBroom(X):- quidditchPlayer(X).

How does Prolog respond to the following queries?

1. wizard(ron).
2. witch(ron).
3. wizard(hermione).
4. witch(hermione).
5. wizard(harry).
6. wizard(Y).
7. witch(Y).

Exercise 4: word(determiner,a).

word(determiner,every).
word(noun,criminal).
word(noun,'big kahuna burger').
word(verb,eats).
word(verb,likes).

sentence(Word1,Word2,Word3,Word4,Word5):-

word(determiner,Word1),
word(noun,Word2),
word(verb,Word3),
word(determiner,Word4),
word(noun,Word5).

What query do you have to pose in order to find out which sentences the grammar can generate? List all sentences that this grammar can generate in the order that Prolog will generate them in.

Exercise 5 Suppose we are given a knowledge base with the following facts:

```
tran(eins,one).
tran(zwei,two).
tran(drei,three).
tran(vier,four).
tran(fuenf,five).
tran(sechs,six).
tran(sieben,seven).
tran(acht,eight).
tran(neun,nine).
```

Write a predicate `listtran(G,E)` which translates a list of German number words to the corresponding list of English number words. For example:

```
listtran([eins,neun,zwei],X).
```

should give:

```
X = [one,nine,two].
```

Your program should also work in the other direction. For example, if you give it the query

```
listtran(X,[one,seven,six,two]).
```

it should return:

```
X = [eins,sieben,sechs,zwei].
```

Exercise 6 Write a predicate `addone/2` whose first argument is a list of integers, and whose second argument is the list of integers obtained by adding 1 to each integer in the first list. For example, the query

```
?- addone([1,2,7,2],X).
should give
X = [2,3,8,3].
```

Topic Database Manipulation

Exercise 7 Suppose we start with an empty database. We then give the command:

```
assert(q(a,b)), assertz(q(1,2)), asserta(q(foo,blug)).
```

What does the database now contain?

We then give the command:

```
retract(q(1,2)), assertz( (p(X) :- h(X)) ).
```

What does the database now contain?

We then give the command:

```
retractall(q(_,_)).
```

What does the database now contain?

Topic Prolog Advantages and Disadvantages

Advantages :

1. Easy to build database. Doesn't need a lot of programming effort.
2. Pattern matching is easy. Search is recursion based.
3. It has built in list handling. Makes it easier to play with any algorithm involving lists.

Disadvantages :

1. LISP (another logic programming language) dominates over prolog with respect to I/O features.
2. Sometimes input and output is not easy.

8) Describe the Prolog search strategy

9)Find answer for given query:

```
friend(jin, james).
```

```
friend(jin, john).
```

```
likes(john, jin).
```

```
likes(james, john).
```

```
happy(X):-friend(X,Y),likes(Y,X).
```

Query : Who is happy?

10) Explain the unification and resolution in prolog

11) Explain what are facts, rules, and queries in logic programming with example.

12) **Draw Resolution graph**

charlie studies csc135

olivia studies csc135

jack studies csc131

arthur studies csc134

kirke teaches csc135

collins teaches csc131

collins teaches csc171

juniper teaches csc134

X is a professor of Y if X teaches C and Y studies C.

Queries / Goals & answers: 1) studies(charlie, What).

2) professor(kirke, Students).