

# Car Maintenance Expert System

Samerah Mohammed Alatawi – 421004747

Boudor Meshal Alrashedi – 431008184

## 1. Introduction

Vehicle maintenance is a very important phenomenon of vehicle ownership through which cars become safer, reliable, and long-lasting. As automotive technology has significantly progressed in the current world, the newer vehicles are sophisticated with high caliber electronic control units, sensors, and mechanical systems. This has come with improvements which in addition have made it even hard to know and solve the problems when it comes.

In the past, troubleshooting of cars has been greatly dependent on the skill and expertise of mechanics themselves. It is not always time and resource-efficient, however, or even consistent or available it may only be accessed. To overcome these short comings, expert systems seem like a intelligent and efficient way to solve the problem because they simulate the human thought process by using rules and facts.

This project entails the work of a very simple expert system in the area of car maintenance implemented in Prolog which is a logic programming language that has specific support to knowledge based systems. The system could be helpful by asking questions about symptoms, processing the information and giving a probable cause or proposed corrective advice. It can be considered as a working example of the use of artificial intelligence to assist with the decision-making in real-life applications concerning such issues as the automotive diagnostics.

## 2. Knowledge Sources

Several sources of authentic materials were used to develop the knowledge behind this expert system and they include:

- **Automotive Repair Manuals:** Reference books on problems and solutions on all kinds and models of cars, presenting general symptoms and customary take-care-of.
- **Manufacturer Specifications:** Those details supplied by the manufactures of the cars themselves on maintenance, fluids and parts.
- **Pro Mechanic Rants:** Information and wisdom by professional mechanics on common problems and how to fix those problems step by step.

- **Online Automotive Forums:** Forums where the car owners interact and exchange their experiences and solutions to different car problems in line with community support.

### 3. Facts

The following predicates represent the facts used in the expert system. They are organized into related groups:

%Fact

% Symptoms

symptom(engine\_no\_start).

symptom(rough\_idle).

symptom(brake\_warning\_light).

symptom(squeaking\_brakes).

symptom(low\_oil\_pressure).

symptom(overheating).

symptom(flat\_tire).

symptom(battery\_dead).

% car issues

issue(battery\_issue).

issue(engine\_issue).

issue(brake\_issue).

issue(oil\_issue).

issue(tire\_issue).

issue(cooling\_system\_issue).

```
% Issue symptoms

has_symptom(battery_issue, battery_dead).

has_symptom(engine_issue, engine_no_start).

has_symptom(engine_issue, rough_idle).

has_symptom(brake_issue, brake_warning_light).

has_symptom(brake_issue, squeaking_brakes).

has_symptom(oil_issue, low_oil_pressure).

has_symptom(cooling_system_issue, overheating).

has_symptom(tire_issue, flat_tire).
```

## 4. Rules

The following rules are used to infer car issues based on the symptoms presented by the user. They are organized into related groups:

```
% Rule to diagnose

diagnose(Issue) :-
    has_symptom(Issue, battery_dead),
    Issue = battery_issue.

diagnose(Issue) :-
    has_symptom(Issue, engine_no_start),
    Issue = engine_issue.
```

```
diagnose(Issue) :-
```

```
    has_symptom(Issue, rough_idle),
```

```
    Issue = engine_issue.
```

```
diagnose(Issue) :-
```

```
    has_symptom(Issue, brake_warning_light),
```

```
    Issue = brake_issue.
```

```
diagnose(Issue) :-
```

```
    has_symptom(Issue, squeaking_brakes),
```

```
    Issue = brake_issue.
```

```
diagnose(Issue) :-
```

```
    has_symptom(Issue, low_oil_pressure),
```

```
    Issue = oil_issue.
```

```
diagnose(Issue) :-
```

```
    has_symptom(Issue, overheating),
```

```
    Issue = cooling_system_issue.
```

```
diagnose(Issue) :-
```

```
    has_symptom(Issue, flat_tire),
```

```
    Issue = tire_issue.
```

## 5. The Prolog Source File

Here is the complete Prolog source code for the expert system, saved in a file named `car_maintenance_system.pl`:

```
%Fact

% Symptoms

symptom(engine_no_start).

symptom(rough_idle).

symptom(brake_warning_light).

symptom(squeaking_brakes).

symptom(low_oil_pressure).

symptom(overheating).

symptom(flat_tire).

symptom(battery_dead).


% car issues

issue(battery_issue).

issue(engine_issue).

issue(brake_issue).

issue(oil_issue).

issue(tire_issue).

issue(cooling_system_issue).


% Issue symptoms

has_symptom(battery_issue, battery_dead).
```

```
has_symptom(engine_issue, engine_no_start).  
has_symptom(engine_issue, rough_idle).  
has_symptom(brake_issue, brake_warning_light).  
has_symptom(brake_issue, squeaking_brakes).  
has_symptom(oil_issue, low_oil_pressure).  
has_symptom(cooling_system_issue, overheating).  
has_symptom(tire_issue, flat_tire).
```

```
% Rule to diagnose
```

```
diagnose(Issue) :-
```

```
    has_symptom(Issue, battery_dead),
```

```
    Issue = battery_issue.
```

```
diagnose(Issue) :-
```

```
    has_symptom(Issue, engine_no_start),
```

```
    Issue = engine_issue.
```

```
diagnose(Issue) :-
```

```
    has_symptom(Issue, rough_idle),
```

```
    Issue = engine_issue.
```

```
diagnose(Issue) :-
```

```
    has_symptom(Issue, brake_warning_light),
```

```
    Issue = brake_issue.
```

```
diagnose(Issue) :-
```

```
    has_symptom(Issue, squeaking_brakes),
```

```
Issue = brake_issue.
```

```
diagnose(Issue) :-
```

```
    has_symptom(Issue, low_oil_pressure),
```

```
    Issue = oil_issue.
```

```
diagnose(Issue) :-
```

```
    has_symptom(Issue, overheating),
```

```
    Issue = cooling_system_issue.
```

```
diagnose(Issue) :-
```

```
    has_symptom(Issue, flat_tire),
```

```
    Issue = tire_issue.
```

```
:- dynamic observed/1.
```

```
start :-
```

```
    write('Car Maintenance Expert System'), nl,
```

```
    write('Enter symptoms (end with "done."): '), nl,
```

```
    read_symptoms,
```

```
    findall(Issue, (diagnose(Issue), has_observed_symptom(Issue)),  
Issues),
```

```
    print_issues(Issues),
```

```
    retractall(observed(_)).
```

```
read_symptoms :-
```



```
repeat,  
write('Symptom: '),  
read(Symptom),  
(Symptom == done -> ! ; assertz(observed(Symptom)), fail).  
  
has_observed_symptom(Issue) :-  
    has_symptom(Issue, Symptom),  
    observed(Symptom).  
  
print_issues([]) :-  
    write('No issues diagnosed based on the entered symptoms.'), nl.  
print_issues([H|T]) :-  
    write('Possible issue: '), write(H), nl,  
    print_issues(T).
```

This expert system can be expanded with additional car issues and symptoms as needed making it a versatile tool for car maintenance guidance