PROJECT DESCRIPTION

1. INTRODUCTION

Instead of learning big methodologies [...] focus on practices. [...] You can become an expert in a practice without being an expert in a complete methodology.
— Ivar Jacobson

This document describes the work towards the project to be done and delivered, as well as any constraints involved in doing so.

The work towards the project is divided into deliverables and each deliverable is divided into **TODOs**.

2. GENERAL

It's much easier to make something more complicated than easy, since people are more impressed by complexity than simplicity.
— Joseph C. Giarratano

The following TODOs apply to **each** of the project deliverables D1, D2, and D3.

Each deliverable will require **submission and demonstration**.

GENERAL TODO 1

Get an account on **GitHub** by the set deadline. Having such an account has a number of advantages, including the following:

- You benefit from the advantages of cloud computing.
- Your work on the project is under version control.
- You develop a habit of working on a deliverable iteratively and incrementally.
- You can prove that the work on the project is from scratch, entirely of your own.

Share the specifics of your account with the teaching assistants. Use your GitHub space for archiving all work on the project, including on the deliverables. Use a **proper naming scheme** for directories and files. Include the specifics of your account in **all** your submissions so that those submissions can be accessed, read, and processed independently, at any time, by the teaching assistants.

GENERAL TODO 2

Make sure that all non-original work (that is, any work not by you) in D1, D2, and D3 is **cited and referenced appropriately**. A comprehensive collection of resources on citing and referencing is available[1].

This includes the use of any **generative artificial intelligence (GAI)** tools, an **appropriate use** of which is encouraged. In case you use any part of the output from a prompt to a GAI tool, include the name of the tool, a copy of the prompt along with the corresponding output, and the timestamp.

GENERAL TODO 3

Make sure that all expression in any form is **evidence-based**, and **to not make claims that cannot be substantiated**. Furthermore, **do not copy others' work verbatim** regardless of whether it is cited. A copy could be **syntactic, semantic, and/or pragmatic**. A copied work does not receive any credit, as the credit goes to the original, not to its copy.

GENERAL TODO 4

Document the deliverable in (1) **Microsoft PowerPoint**, for which a template has been provided, or (2) **LATEX report class**, for which a template could be found on the Web (such as at Overleaf). If (2), then submit both the LATEX source and corresponding PDF.

GENERAL TODO 5

Make sure that the deliverable is **doable** by the **deadline**.

---

[1] URL: http://library.concordia.ca/help/howto/citations.html .

3. DELIVERABLE 1 (D1)

DELIVERABLE 1 TODO 1 [10 MARKS]

Find a **problem domain**, and a subset of it, the **knowledge domain**, D. Give a description of D. D must satisfy the following:

- D is such that (1) you believe you are an expert in it or are interested in becoming an expert in it, or (2) you know someone who is an expert in it, or **both** (1) and (2).
- D has **authoritative information** available publicly, should you need to consult it independently on your own. For example, such information could be in the form of peer-reviewed **articles, books, standards**, and so on.
- D is such that your **experiential knowledge** of it can be **useful** to **many** others. For example, such experiential knowledge could be about how to **diagnose something**, how to **decide something**, or how to **do something** as well as how to **not do something**.

DELIVERABLE 1 TODO 2 [10 MARKS]

State the **goal**, G, for your project. G must satisfy the following:

- G must be based on D.
- G should aim to be **SMART**.

DELIVERABLE 1 TODO 3 [10 MARKS]

Find a potential **user**, U, of your expert system. Give a description of U.

DELIVERABLE 1 TODO 4 [10 + 30 + 30 + 20 MARKS]

Develop a **factbase**, F, for D. Develop a **rulebase**, R, for D. Let K be the **combination** of F and R. K must satisfy the following:

- **Requirements:**
  - K helps U towards G.
  - K aims to be explainable, maintainable, portable, readable, reusable, understandable, and usable.
  - K has a non-trivial size. (For example, F has at least 10 useful facts about D and R has at least 20 useful rules about D.)
- **Design:**
  - K is expressed in Structured English for syntax and uses propositional logic or predicate logic, as applicable, for semantics.
  - K uses a proper naming scheme. (For example, concepts, facts, and rules use the natural naming scheme.)
  - K has a balance of granularity. (For example, the rules are neither too small, nor too large.)

- **Implementation:**
  - K is represented in the declarative programming paradigm of the CLIPS programming language.
  - K has support for debugging.
  - K uses a proper programming style. (For example, it uses a proper naming scheme; uses comments in appropriate places; has proper formatting; uses whitespace where necessary; and so on.)
  - K is modular. (For example, F groups similar facts and F groups similar rules.)
  - K separates facts and rules. (For example, K has a text file for F and a text file for R for use by the CLIPS IDE and the CLIPS inference engine.)
- **Validation and Verification:**
  - K is reviewed and tested, symmetrically or asymmetrically wherever possible. (The "symmetric review" applies only to those students who have opted for Option B. The "asymmetric review" applies only to those students who have opted for Option C.)