

CS1203

Assignment 7

Isaac Shoebottom

3429069

“I warrant that this is my own work”

1. All three types of programs, an assembler, a compiler, and an interpreter all do the same thing in essence. They take code given and produce an output that a computer can use to execute the code, because computers cannot read code not written in machine language.

- 1) An assembler could be called a very specific kind of compiler. An assembler does not need to do as much to generate machine code. Assembly is often inherently close to the machine language instructions for the given processor, and therefore the assembler just needs to translate the assembly instructions to a machine code executable, in a format the operating system can read.
- 2) A compiler is a program that can read a raw text file, figure out what the text written inside means, due to the formatting of the text and uses the code to generate a program that the operating system can execute, often all ahead of the time the program is run. Compilers all do the same thing an assembler does, just with a higher-level language that is more readable to humans. There is just in time compilers as well, that name is a kind of a misnomer as just in time compilation is a mix between a compiler and an interpreter.
- 3) An interpreter is a program that reads the text inside the source code file and generates the machine code and runs it at the same time. It does not normally save the output of the machine code as it is being generated. For most applications the downside of generating the machine code every time the program is ran is often negligible as modern computers are fast and the portability of the program is more important. Another downside is that since machine code needs to be generated at runtime, every computer that wishes to run the program must have the interpreter installed, while programs compiled ahead of time are faster and only the developers of the program need to have the compilers installed.

2.

The screenshot shows the SWISH Prolog IDE interface. The top menu bar includes "File", "Edit", "Examples", and "Help". The top right corner shows "346 users online", a search bar, and a notification bell. The main editor area is titled "Example 1.1" and contains the following Prolog code:

```
1 bigger(elephant, horse).
2 bigger(horse, donkey).
3 bigger(donkey, dog).
4 bigger(donkey, monkey).
```

The console on the right displays the results of running the code:

```
bigger(donkey, dog)
true
bigger(elephant, monkey)
false
?- bigger(elephant, monkey)
```

At the bottom right, there are buttons for "Examples", "History", "Solutions", and a "Run" button with a "table results" checkbox.

The screenshot shows the SWISH Prolog IDE interface. The top menu bar includes "File", "Edit", "Examples", and "Help". The top right corner shows "345 users online", a search bar, and a notification bell. The main editor area is titled "Example 1.1" and contains the following Prolog code:

```
1 bigger(elephant, horse).
2 bigger(horse, donkey).
3 bigger(donkey, dog).
4 bigger(donkey, monkey).
5
6 is_bigger(X, Y) :- bigger(X, Y).
7 is_bigger(X, Y) :- bigger(X, Z), is_bigger(Z, Y).
```

The console on the right displays the results of running the code:

```
is_bigger(elephant, monkey).
true
Next 10 100 1,000 Stop
is_bigger(X, donkey).
X = horse
X = elephant
Next 10 100 1,000 Stop
is_bigger(donkey, X), is_bigger(X, monkey).
false
?- is_bigger(donkey, X), is_bigger(X, monkey).
```

At the bottom right, there are buttons for "Examples", "History", "Solutions", and a "Run" button with a "table results" checkbox.