Exercise 5.1: Name Binding

Consider the following program:

1. procedure Namebinding is
2. X, Y : Integer;
3. procedure P1 (Y : Integer) is
4. begin
5. Put (X + Y);
6. end P1;
7. procedure P2 (Y : Integer) is
8. X : Integer;
9. begin
10. X := Y;
11. P1 (Y);
12. end P2;
13. begin
14. X := 0; Y := 2;
15. P1 (Y); P2 (Y);
16. end;

a. What is the output of the program using
   (i) static name binding?
   (ii) dynamic name binding?

b. What is the scope and the lifetime of every variable X using
   (i) static name binding?
   (ii) dynamic name binding?

c. What are the disadvantages of dynamic name binding compared to static name binding?

Exercise 5.2: Descriptors for Name Binding

For which concepts found in programming languages are runtime descriptors required for name binding? For which concepts can they be avoided?
Exercise 5.3: Name Binding and Method Calls

The following source code is written in an object-oriented language.

class T {
    ...
    void m (T x) { print ("T"); }
};

class NT extends T { // inheritance
    ...
    void m (T x) { print ("NT"); }; // redefinition!
    void m2 (NT x) { ... }
};

T ot;
NT ont = new NT();
ot = new NT();
...
ot.m (ot); // (1)
ot.m2 (ont); // (2)
ont.m (ot); // (3)
ont.m2 (ont); // (4)

a. Assume that the language uses dynamic name binding for method calls (like Smalltalk).
   Are the calls at (1) to (4) legal calls? If so, what output will be produced by each call (assume that the calls are alternative implementations, so they do not influence each other)? If not, what kind of error diagnostic do you expect?

b. In the following, the language uses static name binding for method calls (like C++, Java, Eiffel). Answer the questions regarding the legality and output of the calls as in the previous part. Which results are the same? Which differ? Please state the reason for the observed behaviour.

c. What are the advantages of static name binding for method calls?

d. General question: Explain how a method name is bound to a specific method declaration and how the correct body of the method is found for languages using static name binding.

Exercise 5.4: Overloading

a. Give examples of overloading in programming languages.

b. Explain the difference between overloading and overriding?

c. Does overloading cause additional overhead during runtime?

d. Is there a method to statically resolve overloading in the compiler.