**Chapter 5 Homework Solutions**

**MTH 135**

*Review Questions:*

2. The potential danger of case-sensitive names is lack of readability because names that look very similar in fact denote different entities. Case Sensitivity violates the design principle that language constructs that look similar should have similar meanings.

3. Reserved words are better than keywords because the ability to redefine keywords can lead to readability problems.

4. An alias is given to names that each can be used to access the same memory location.

6. The l-value of a variable is its address. The r-value of a variable is its value.

7. A binding is an association, such as between an attribute and an entity or between an operation and a symbol. The time at which a binding takes place is called binding time.

8. The four times (besides language design and language implementation) that a binding can take place in a program are: compile time, load time, link time, or run time.

9. A static binding is one in which the binding first occurs before run time and remains unchanged throughout program execution. If the binding first occurs during run time or can change in the course of program execution, the binding is called dynamic.

12. Static variables are those that are bound to memory cells before program execution begins and remain bound to those same memory cells until program execution terminates. Advantages are: history sensitive variables (retain values between separate executions) and efficiency (direct addressing). Disadvantages are: reduced flexibility (no recursion) and no shared storage among variables.

Stack-dynamic variables are those whose storage bindings are created when their declaration statements are elaborated (run time), but whose types are statically bound. Advantages are: allows recursion and conserves storage. Disadvantages are: overhead of allocation and deallocation, subprograms cannot be history sensitive, and inefficient references (indirect addressing).

Explicit heap-dynamic variables are nameless (abstract) memory cells that are allocated and deallocated by explicit run-time instructions specified by the programmer. Advantage is: provides for dynamic storage management. Disadvantages are: inefficient and unreliable.

Implicit heap-dynamic variables are bound to heap storage only when they are assigned values. In fact all their attributes are bound every time they are assigned. They are names that adapt to whatever use they are asked to serve. Advantage is flexibility. Disadvantages are: inefficient because all attributes are dynamic and loss of error detection.

16. The lifetime of a variable is the time during which the variable is bound to a specific memory location. The scope of a variable is the range of statements in which the variable is visible (can be used). Static scooping is when the scope of a variable can be statically determined, that is, prior to execution. Dynamic scooping is based on the calling sequence of subprograms, not on their spatial relationship to each other.

23. Advantages of named constants are readability, reliability, and ease of maintaining programs. Readability is improved because instead of what appears to be a random number is instead a name that can give the reader an idea of what the number represents. In terms of maintaining programs (aide to reliability), if a constant appears more than once in a program, making it a named constant makes it much easier to change the constant if you ever need to update the program – you only need to change it once instead of every place it appears.

*Problem Set:*

1. “sum\_of\_sales” is most readable because it looks the most like written English does. The underscore acts as the space between the words making it quicker to read that variable.

2. The advantage of a typeless language is flexibility; any variable can be used for any type values. The disadvantage is poor reliability due to the ease with which type errors can be made, coupled with the impossibility of type checking detecting them.

4. Implicit heap-dynamic variables acquire types only when assigned values, which must be at runtime. Therefore, these variables are always dynamically bound to types.

5. Suppose that a Fortran subroutine is used to implement a data structure as an abstraction. In this situation, it is essential that the structure persist between calls to the managing subroutine.

8. Variable Where Declared

In sub1:

a sub1

y sub1

z sub1

x main

In sub2:

a sub2

b sub2

z sub2

y sub1

x main

In sub3:

a sub3

x sub3

w sub3

y main

z main

9. Static scoping: x = 5.

Dynamic scoping: x = 10

10.Point 1: a 1

b 2

c 2

d 2

Point 2: a 1

b 2

c 3

d 3

e 3

Point 3: same as Point 1

Point 4: a 1

b 1

c 1

12. Variable Where Declared

(a) a, x, w sub3

b, z sub2

y sub1

(c) a, y, z sub1

x, w sub3

b sub2

(f) a, y, z sub1

b sub2

x, w sub3

*Programming Exercises:*

5. # include <iostream>

using namespace std;

int main()

{

x = 21;

int x;

x = 42;

cout << "x = " << x;

return 0;

}

It compiled, but here’s what happened when I tried to run it…

./mth135test: extern: command not found

./mth135test: typedef: command not found

./mth135test: typedef: command not found

./mth135test: typedef: command not found

./mth135test: typedef: command not found

./mth135test: typedef: command not found

./mth135test: typedef: command not found

./mth135test: line 46: syntax error near unexpected token `;'

./mth135test: line 46: ` } \_\_quad\_t;'

because I tried to use x (in the assignment x = 21) before I had declared it, I got several type definition errors