**Assignment 3**

**Chapters 5 and 6**

**Chapter 5:**

1. Variables are made up of 6 attributes list and briefly describe what each one tells us about a variable.

The six attributes are name, address, value, type, lifetime, and scope, which name tells us the most common names in a program. The address of a variable is the machine memory address with which it is associated. The value of variable is the contents of the memory cell or cells associated with the variables. The type of variable determines the range of values the variable can store and the set of operations that are defined for values of the type. The lifetime of variable is the time during which the variable is bound to a specific memory location. The scope of variable is the range of statements in which the variable is scope.

2. Describe what an alias is?

Aliases are two or more variables bound to the same storage address. They are regarded as detrimental to reliability but are difficult to eliminate entirely from a language.

3. What is the difference between static binding and dynamic binding.

The binding which can be resolved at compile time by compiler is known as static binding, in which binding of all the static, private and final methods is done at compile-time. In Dynamic binding, compiler doesn’t decide the method to be called. In overriding, both parent and child classes have same method is an example of dynamic binding. Furthermore, the static binding uses type information for binding while dynamic binding uses objects to resolve binding.

4. What are the different times binding can take place?

Language design time —  bind operator symbols to operations

Language implementation time– bind floating point type to a representation

Compile time — bind a variable to a type in C or Java

Load time — bind a C or C++ static variable to a memory cell)

Runtime — bind a non-static local variable to a memory cell

5. What are the advantages and disadvantages of dynamic type binding?

The advantage of the dynamic type binding is it provides more flexibility than static binding., but the disadvantages are the programs with dynamic binding is it easier to get an errors because the error-detection capability pf a compiler is reduces, and during the execution time, the cost of implementation is more expensive.

6. Define static, stack-dynamic, explicit heap-dynamic and implicit heap-dynamic variables.

Static: bound to memory cells before execution begins and remains bound to the same memory cell throughout the execution.

Stack-dynamic: storage bindings are created for variables when their declaration statements are elaborated.

Explicit heap-dynamic: allocated and deallocated by explicit directives, specified by the programmer, which take effect during execution.

Implicit heap-dynamic variables: Allocation and deallocation caused by assignment statements.

7. What is the referencing environment of a statement?

The referencing environment of a statement is a set of all names visible to that statement.

8. Describe how lifetime scope are similar and how they different.

They both defined the region of a program. The differences are the life time of any variable is the time for which the particular variable outlives in memory during running of the program, but the scope of any variable is actually a subset of life time. A variable may be in the memory but may not be accessible though. So, the area of our program where we can actually access our entity is the scope of that variable.

9. Given the following JavaScript Program:

var x;

function sub1(){

document.write(“x = “ + x + “ ”);

}

function sub2(){

var x;

x = 10;

sub1();

}

x = 5;

sub2();

a. What would be outputted under static-scoping rules? X=5

b. What would be outputted under dynamic-scoping rules? x=10

10. Consider the following C program:

void fun (void){

int a, b, c; /\* definition 1 \*/

. . .

while(. . .){

int b, c, d; /\* definition 2 \*/

. . . <---------------------- 1

while (. . .){

int c, d, e; /\* definition 3 \*/

. . . <---------------------- 2

}

. . . <---------------------- 3

}

. . . <---------------------- 4

}

For each of the four marked points in this function, list each visible variable, along with the number of the definition statement that defines it.

1.

Visible variable number of the definition statement that defines it

b,c,d definition 2

a definition 1

2. c,d,e definition 3

b definition 2

a definition 1

3. b,c,d definition 2

a definition 1

4. a,b,c definition 1

**Chapter 6:**

1. What is a descriptor?

A structure containing information that describes data.

1. Define static, fixed stack-dynamic, fixed heap-dynamic, and heap-dynamic arrays. What are the advantages of each?

Static: subscript ranges are statically bound and storage allocation is static (before run-time), the advantage include efficiency, because it does not use dynamic allocation.  
  
Fixed stack-dynamic: subscript ranges are statically bound, but the allocation is done at declaration time. The advantage includes space efficiency.  
  
Stack-dynamic: subscript ranges are dynamically bound and the storage allocation is dynamic (done at run-time), the advantage is flexibility because the size of an array does not required to be known.

Fixed heap-dynamic: similar to fixed stack-dynamic which storage binding is dynamic but fixed after allocation, the advantage includes space efficiency, because binding is done in heap bot stack.

Heap-dynamic: binding of subscript ranges and storage allocation is dynamic and can change any number of times. The advantage is flexibility because array can grow or shrink during execution time.

1. Describe what a tuple is and how it **differs from a record**.

A tuple is an ordered group of elements.

A record is typically a group of named elements such that x:10, y:20 where the value has two fields labelled x and y and the value of field x is 10.

The tuple and record are both product types which allow build types from multiple simpler types.

1. What are the two common problems with pointers?

The first common problem is dangling problem which a pointer that contains the address of a heap-dynamic variable that has been deallocated. The second common problem is causing lost heap-dynamic variable which is an allocated heap-dynamic variable that is not longer accessible to the user program.

1. What advantages do Java and C# reference type variables have over the pointers in other languages?

The reference Java and C# provide some of the flexibility and the capabilities of pointers, without the hazards.

1. Describe the lazy and eager approaches to reclaiming garbage.

Eager—maintain a reference count in every block, update it whenever a pointer to block changes.

Lazy--- When no more available memory blocks, garbage collect by tracking every pointer into heap and ‘marking’ the block as used.

1. What is type error?

A type error should thrown when a value is a different type than what was expected.

1. What is a strongly typed?

Strongly typed when every type error is detected.

1. What is type coercion and what languages do not have any type coercions.

Type coercion is a process that convert a value of a type to another type.

F# does not have type coercions.

1. What is the difference between name type equivalence and structure type equivalence.

Under name type equivalence, two variables have the same type if they are defined in the same declaration or in declarations that use the same type name.

Under structure type equivalence, two variables have the same type if they have identical structures.