Assignmnet-6

1. **Some programming languages are type-less. What are the obvious advantages and disadvantages of having no types in a language?**

Advantages:

- It is easy for programmer that not need to figure out nor keep track of the type of each variable.
- Easier to learn
- Easier to change in a program (example changing a variable type in a program)
- It makes it more flexible.
- Errors will show up at run time when built-in functions are used with arguments that don't make sense.

Disadvantages:

- Managing memory in runtime. As there's no visibility and a blindfolded scenario.
- It is easier to make mistakes and to confuse variables.
- Example: if we have variable var test = 10 and then we assign test = false and user try to attempt test = test + 10; This will create an error.

2. Write a simple assignment statement with one arithmetic operator in some language you know. For each component of the statement, list the various bindings that are required to determine the semantics when the statement is executed. For each binding, indicate the binding time used for the language.

Programming language: **JAVA**
int i = 0;
i = i + 1;
Variable type : Compile time binding
Operator '=' : Design time binding
Operator '+' : Compile time binding
Value of 'total' : Run time binding

3. Dynamic type binding is closely related to implicit heap-dynamic variables. Explain this relationship.

Dynamic binding is basically resolving binding at runtime. Whereas by definition, implicit heap dynamic variables are bound to be a type at runtime when the value gets assigned to the variable at runtime, hence both dynamic binding and implicit heap dynamic variables are closely related.

4. Describe a situation when a history-sensitive variable in a subprogram is useful.

History-sensitive variables are used in a situation, where some operation is performed on a variable, and the function exits, then the function is called again. This variable can retain its value between different function calls. This way, the function doesn't have to take the variable as a parameter, but only to return it. Example: - Pseudorandom number generator.

5. What is the general problem with static scoping?

Static scoping gives you more access to variables and subprograms than you like. Regardless of when it is executed in runtime, the variables are bound at compile time. The issue is to program evolution. However, since our programs are highly dynamic, they are used to changing on a regular basis, so static variables must also change. These changes often result in restructuring, thereby destroying the initial structure that restricted variable and subprogram access.