Compiler Construction

(BSCS – 601)

Keywords: (Same as C#)

- ret (return)
- class
- new
- static
- interface
- abstract
- break/continue
- closed //closed represents sealed
- virtual
- Override

Data Types:

- char

RE=”^’[.]{1,2}’$”

- Int

RE= ”^-?[0-9]{1,16}$”

- str (string)

RE=”^\”[\\s\\w.]*\”$”

- bool

RE = “^(true|false)$”

- double

RE = “^-?[0-9]+(.[0-9]+)?(e)?(-)([0-9]+)?$”

Data Structure:

- Array
- ArrayList

Loops:

- for : initialization ; condition ; Increment/Decrement : (body)
- foreach : object [var] in collection : (body)
- while : condition : (body)

Syntax:

```
        for: int i =0, i < 3, i++:
            (
                    Console.Write(“Hello ”);
                    Console.WriteLine(“World”);
            )
```

Operators: (Same as C#)

- Arithmetic Operators: +, -, *, /, %
- PM : +, -
- PMD : *, / , %
- Relational Operators: <, >, <=, >=, ==, !=
- Logical Operators: &&, ||
- Unary Operator : !
- Assignment Operators: =, +=, -=, /=, %=
- Increment/Decrement Operators: ++, --

Conditions:

- if : condition : (body)
- else : (body)
- elif : condition : (body)
  //elif is for else if

Access Modifiers: (Same as C#)

- public
- private
- protected

Functions:

- data type : parameters : (statement)
  Syntax:

```
        // no Paramter
        Int Increase:
        (
                num = num + 1;
                return num;
        )

        // Multiply Paramter
        Int Area: int width, int height:
        (
                int area = width * height;
                Return area;
        )
```

- void : parameters : (statement;)

Syntax:

```
    void Display:
    (
        Console.WriteLine(“Hello World”);
    )
```

OOP Concepts:

- Inheritance represents as colon (:)
- Interface represents as colon (:) or comma (,) if there is any inheritance(like C#)
- VO: override and virtual

Terminator (End of Line):

- Semicolon (;)