-> class hierarchy String->Object->List->Char. Type constructor

< left angle bracket. Used for <type>

> rtight angle bracket. Used for <type>

, dereferencing, For an object Object, Object.run() runs a function

entitypool main memory pool

stream stream n = newTypeName (initializes streams)

pool class retrieves memory for a class from the entityPool

( ) initialize variable with a stream to put on

TypeName (n) variablename, where n is the stream name

.get n.get(variablename) returns the value of that variable on that stream

.set n.set(variablename) sets the value of that variable on the stream

< Instantiating classes. <TypeName> ClassName myObject = new <>();

> No need for redundancy in instantiating a class, Calls default constructor

Type names are placed within brackets

= Asssignment, non mathematical (assignment of objects)

<> Anonymous TypeName

() Instantiation of a class (empty constructor)

( Used to calll a function or constructor

) Used to call a function or constructor

; Line terminator

volatile Can be changed by a background process

void

short

char

Long

boolean

int

double

float

fixed

uInt

uShort

uLong

enum

collection

set

array

arraylist

linkedlist

doublylinkedlist

doublylinkedarraylist

vector

hashtable

map

multimap

string

object

list

tree

stack

queue

extends Declares this class as a subclass of one or more classes

Implements Declares this class as an implementation of an interface

let Used to createtypeless variables

dynamic

release

retain

autorelease

garbage off

garbage on

retain similar to new and delete in c++

release

frozen CASE is by default dynamically typed. You could do: int myInt;

Then later myInt = “My Int”. frozen makes a var in CASE

Statically typed.

thaw Turns off static typing

lock Used with threads and mutexes

synchronized Used with threads

[ : ] Message Passing from source to destination

printbug Outputs a debugging statement

assert Throws an error if an item is false

[ ] Method brackets

try

catch

last

await Used in try….catch

autofall Used in try...catch

tag Used in try….catch so t hat you don’t have to initialize to null

tefore a try catch

throw Throw an exception

throws This class throws an exception

( ) Used for typecasting

struct

union

sealed Prevents a class from inheriting from the sealed class

this Access member variables and functions of the class

inline Inserts assembly code into the CASE code

bytecode Like inline, but for bytecode

null Discouraged. Represents a null reference

as If we want to give a module a different reference

lambda Creates a new anonymous function

global Access variable outside functions

del Deletes objects

pass Does nothing

command execute Executes a command on the command line and returns as a

String

true boolean true

false boolean false

sizeof size in bytes

return exits a function

register loads a value into a CPU register

extern states that an element is in another file

continue Passes control to the beginning of the loop

namespace Creates a scope for code. Two functions can exist at the same

time in different namespaces

@texts: A text string literal

“ “ An empty string

“ String declaration

transient Declares that this element is not part of the serialization of an

Object

auto Declares that a variable only has scope local to the outermost

static For functions declares only class scope, For local variables

retains value for every execution,

final Assigns a value to a variable. Can only be performed once.

( c ) String concatenation.

( m ) String subtraction (removes a substring)

?? Checks to see if an object is initialized

native Use code from other languages

requires Makes it so anybody who wants to use this code must use

The keyword satisfies

satisfies

package Defines a group of class files that are importable

import Imports a package into a class file

\* String wildcards. Can be used for calling functions and

Referencing variables

^ Exponentiation

- Negate target

+ Opposite of negate

+ Addition

- Subtraction

div Integer division

/ Floating Point division

\* Multiplication

rem Remainder

mod Modulo

% percent sign

(<<) shift left operator

(>>) shift right operator

= arithmetic assignment operator

== boolean equals

< less than operator

> greater than operator

<= less than or equals operator

>= greater than or equals operator

<> not equal to

! factorial

++ post fix or prefix increment operator

-- post fix or prefix decrement operator

+= compound addition

-= compound subtraction

/= compound division

\*= compound multiplication

` automatically evaluate entity

‘ turn the expression into a list

. used to join two brace statements

{ } surrounds blocks of statements (specificallly methods)

Each line of executable code is surrounded by braces e.g.

{ int n = 5 }  
 { int g = 6 }

{ n = g+ 5 }

{return n}

, can be used to join statements within braces,

For example { a=5, print(“Helllo”}

endclass Terminates a class definition

exp Exponential to a power

abs Absolute value

matrix As in Linear Algebra Matrix

vector As in Linear Algebra Vector

in method Checks to see if an object is in a method

in class Checks to see if an object is in a class

convo Turns a string into a list

car Takes a list and peels off the first element

cdr Takes a list and peels off everything but the first element

if

else tpyical else except it works on while loops as well

do

while

where A sequence of instructions that eecutes on the where condition

until

residue Keeps a variable in loop scope after the loop terminates

resolve When using an enhanced loop it is handy to see the iterator

Index.

wait Blocks execution until a certain condition is met

leap Typically in an ehanced loop you can’t modify the control

Variables contentse. Leap allows you to execute such

Commands.

iterator A variable that represents the index in a loop

foreach An enhanced for loop

: Operator used in enhanced loop

; Used to seperate elements of a traditional for loop

for A traditional for loop and used in list comprehensions

[[ ]] Brackets that surround list comprehensions

( ) Used in a simple for loop: for (<50) { Print:”Print 5 times” }

goto DIscouraged. Branch.

switch A switch statement

case A case in a switch statement

exit Passes control out of the current scope

root Passes control completely to the root scope

break Escapes from the switch statement

== {==5.0} { return String(50) }

fall Causes execution to fall to the next case

branch Go to another thread

label A marker for gotio to branch to

jump Go to a function

class Used with inner classes. Class myInnerClass {....}

new Used with anonytmous inner classes. New Class { …. }

new Used to create objects and invoke constructors

new Used with initializing arrays and other primitives

new Used to hide a member from a base class member

range Used to in iterating with a for loop

len Used to see the length of a list

with Temporarily changes the scope we are in

neutral Placeholder for datatypes in instantiation

sequence Defines a sequence. Used as a classname

In Determines if an object contains a value

is Do the two objects point to the same reference

Is not Opposite of Is

Is nothing Opposite of Is

type of R eturns the type of an object

Instance of Returns if an object is an instance of a class

get type Get the type of an object or class

entitypool get instance Gets an instance of a class from a stream

proto { } myClass = Proto { DefaultClass }

.class Can be used instead of doing Class.getClass()

get Set code for a the get property of a class

set Set code for the set property of a class

=> Expression body method operator

{ } Body of code for the expression body method op

interface Declares the header for an interface body

abstract Declares a class or method as abstracft

construct Declares a class constructor for a class

explicit Forces type casting

super Access super class

override Overrides a virtual method

inclusion Code that only executes for the first iteration ofthe loop

Iterator An iterator to enhance for loops

\* Derefences an iterator back to its host class. Gives the value

At that iteration

rename Dynamically rename an object

object Shortcut for the Objecet class

class<?> Used for reflection

|| Multiple statements on one line

Meta MetaClass is a class whose instance is a class

x: Defines a parameter to be a mixin class

[ ] Array indexer

[ ] [ ] [ ]... Multdimensional arrays

[ ] Object indexer

base Used to access the base class of a chain of inheritance

Unlike the super class which only goes up one level

tuple A group of objects similar to an array but of any types

set Like tuples and lists, but one instance of each element and

Unordered,

.union

.intersect

.difference

.symmetric difference

shall GIve access to specific classes, despite public or private

friend Marks a function in a class that can be accessed despite

Public private modifiers,

class{T} Generics. Can pass a classname for the T parameter,

@optional Implementation of method in inheritance is optional

@required Implementation of method in inheritance is required

: caller Specify what functions/classes may call this method

, Used to seperate multiple return types

( , ) Return multiple return values in the form of a tuple

procedure A method that may not return a value

function A method that may return a value

syn One method call with the same signature but does different

Things

$precondition Define preconditions for a function

$postcondition Define postconditions for a function

virtual Marks the method for being able to be overridden

: Compose functions

@ Used for function concatenation

[[ ]] Brackets for rewrite functionality

rewrite Rewrite a piece of code dynamically

endrewrite End the replacement block

write Target a block of code for replacement

endwrite End code block for replacement

delegate Used like function pointeres

\*=> Define a lambda expression

arg\* Allows for any number of input parameters

implicit User defined type conversion

to string Outputs a string from an object

from string initializes the named class from the entity pool

convert Convert the callling class into another class

event.fire FIres an event to the global event queue

trigger Triggers a listener with the same parameter

listener Put in header of function. Executes on fire of same parametert

& TypeName ( n ) bobsPointer = &variableName. Pointers

\* Dereference pointer. (\*myClass).pointer

-> Shortcut. Equal to (\*myClass).pointer

const Keeps a parameter from being modified,

Functionname(const int parameter)

% format specifiers

|  |  |
| --- | --- |
| i | Integer |
| d or u | Decimal integer |
| o | Octal integer |
| x | Hexadecimal integer |
| f, e, g | Floating point number |
| a |
| c | Character |
| s | String of characters |
| p | Pointer address |
| [characters] | Scanset |
| [^characters] | Negated scanset |
| n | Count |
| % | % |

Escape Characters

|  |  |
| --- | --- |
| \t | Insert a tab in the text at this point. |
| \b | Insert a backspace in the text at this point. |
| \n | Insert a newline in the text at this point. |
| \r | Insert a carriage return in the text at this point. |
| \f | Insert a formfeed in the text at this point. |
| \' | Insert a single quote character in the text at this point. |
| \" | Insert a double quote character in the text at this point. |
| \\ | Backslash |
| \A | Alarm (Beep, Bell) |
| \t | Horizontal Tab |
| \v | Vertical Tab |
| \? | Quotation Mark |
| \nnn | The character whose numerical value is given by nnn interpreted as an [octal](http://en.wikipedia.org/wiki/Octal) number |
| \xhh | The character whose numerical value is given by hh interpreted as a [hexadecimal](http://en.wikipedia.org/wiki/Hexadecimal) number |
|  |  |
|  |  |

str Converts an object into a string

[ : ] Slicing (a la Python)

**list** A built-in class name for bracketed,.comma seperated lists

.append Add an item to the end of the list

.extend Extend the list by appending all the itemsin a collection

.insert Insert an item in a given position

.remove Remove an item from a list

.pop rEmove the item at the given position in the list and return it

.clear Remove all items from the list

.index Return an item from the list

.count Counts the number of items in the list

.sort SOrt the items in place

.copy Returns a shallow copy

**tuple** A built in class name for paranthesed, comma seperated lists

<< Extraction operator used like C-style streams

>> Insertion operator used like C-style streams

deep Performs deep copy

need Pass by need

& Pass by reference

operator Used in operator overloading

&= Copy string by content,m not reference

= Perform sdeep copy

~ Perform shallow copy

=$ Treat string copy as the Universal Base Case

public

protected

internal

private

const declare a constant variable

car Take the head of a list and return it

cdr Take everything but the head of a list and return it

cons Like in LISP

base For use in recursion

tail For use in recursion

@[ ] An annotation. Provides data that is not specifically about the program

// Line comments

/\* \*/ Block comments

/\*\*\*\*\*/ Documentation

( ) Generator. looks like a list comprehension but with paranthesis instead of

brackets.

Like a collection but calculates on the fly, doesn’t keep anything in

memory

# Prefix for declaration of a class.

String->Object->MyClass

#public class Main [

[public procedure void main()

{ Print(“Hello World”);}

]

] end class

Creating a variable from the entity pool:

{entitypool.create();}

{stream n = new realNumbersType}

{realNumbersType Irrationals ( n ) pi}

query entity pool Retrieve an anonymous objecvt from the entity pool to manage

Memory more efficiently

stateful Stateful and Tie tie two variables together so that when one changes

The other changes as well. Also, you can tie to an operator or function

So that when the variable changes that function or operator is calledd

tie

equiv One-sided assignment statement

controller Assigns a function as a controller in MVC

view Assigns a function as a view in MVC

model Assigns a function as a model in MVC

% boolean operators

|  |  |
| --- | --- |
| True |  |
| False |  |
| Not |  |
| IsTrue |  |
| IsFalse |  |
| And |  |
| Or |  |
| Xor |  |
| NotOr | Prefixing a word with Not negates it |
| NotAnd |  |
| Like | Pattern Comparison Operator |
| & | Bitwise Operator |
| | | Bitwise Operator |
| \| | Bitwise XOR |
| == | Boolean Equals? |
| Also | Often times in C/C++/Java you have to test two boolean expressions at a time.  This results in a mess of operators that is frustrating to read .Using Also….  If (h==5 also n>5) replaces something like: if ((h==5) && (n>5)) |
|  |  |

|  |  |
| --- | --- |
| boolean isLetter(char ch)  boolean isDigit(char ch) | Determines whether the specified char value is a letter or a digit, respectively. |
| boolean isWhitespace(char ch) | Determines whether the specified char value is white space. |
| boolean isUpperCase(char ch)  boolean isLowerCase(char ch) | Determines whether the specified char value is uppercase or lowercase, respectively. |
| char toUpperCase(char ch)  char toLowerCase(char ch) | Returns the uppercase or lowercase form of the specified char value. |
| toString(char ch) |  |

Like match a string against a pattern or vice-versa