Classes

Declaration

All member attributes and functions are declared and defined into the class definition.

```
class myClass
      // all definitions go here
end class
```

Constructors

A constructor is a kind of member function that initializes an instance of its class. It is executed whenever we create new objects of that class.
In EZ :

- there is no such thing as data encapsulation : all attributes are public by default
- the default constructor and the parameterized constructor do not need to be defined

Syntax

```
c is MyClass
c is MyClass(val1, val2)
c is MyClass(Att1=”val1”, Att3=”val2”)
```

Examples

```
class Person
begin
      nom is string
      prenom is string
end class

// Declaration with default constructor
p1 is Person
// Declaration with parameterized constructor
p2 is Person( "nom" ,  "prenom" )
p3 is Person(nom= "nom" , prenom= "prenom" )
p4 is Person(age= 20 )
```

Destructors

A destructor is a special member function of a class that is executed whenever an object of it's class goes out of scope or whenever the delete expression is applied to a pointer to the object of that class. Destructor can be very useful for releasing resources before coming out of the program like closing files, releasing memories etc.

Example :

```
class MyClass
begin
      destruct
            print "MyClass destructor"
      end destruct
end class
```

Getters

In C++, a getter is a member function of a class that allows access to private data. Since there is no data encapsulation in EZ, getters are useless : they are not implemented.

Setters

In C++, a setter is a member function of a class that allows to modify private data. Since there is no data encapsulation in EZ, setters are useless : they are not implemented.

Operators overloading

You can redefine or overload most of the built-in operators available in EZ. Thus a programmer can use operators with user-defined types as well.
Overloaded operators are functions with special names the keyword operator followed by the symbol for the operator being defined. Like any other function, an overloaded operator has a return type and a parameter list.

Examples

```
operator== (c is C)  return bool
      ...
end
```

```
operator< (c is C)  return bool
      ...
end

operator= (c is C)  return C
      ...
end
```

“print” function overloading

The print function can be overloaded in a class in order to specify how to display its information.

Examples

```
class Person
begin
      nom is string
      prenom is string
      age is integer

      procedure print()
            print “nom: ” , nom, “ prenom:”, prenom, “ age: ”, age
      end procedure
end class

//main
program main

procedure main()
      p is Person(“dupont”,”laurent”,25)
      print p
end procedure

// ouput
// nom: dupont prenom: laurent age: 25
```

Static members of a class

We can define class members static using **shared** keyword. When we declare a member of a class as static it means no matter how many objects of the class are created, there is only one copy of the static member : this member is shared by all objects of the class.

Example

```
class MyClass
begin
      a is shared integer = 0
end class
```

Static Function Members

By declaring a function member as shared, you make it independent of any particular object of the class. A shared member function can be called even if no objects of the class exist and the **shared** functions are accessed using only the class name and the operator .
A shared member function can only access shared data member, other shared member functions and any other functions from outside the class.

Examples

```
class A
begin
      // non static
      var1 is integer
      procedure f1() begin
            ...
      end procedure

      // static
      var2 is shared integer = 0
      shared procedure f2() begin
            ...
      end procedure
end class


// function main
procedure main()
      // non static
      a is A
      a.var1 = 1
      a.f1()

      //static
      A.var2 = 1;
      A.f2();
end procedure
```