Introduction

1. Encapsulation by Subprogram
2. Generic Subprogram

Encapsulation by Subprogram

A subprogram is a n abstract operation defined by the programmer. A subprogram have to views:

- Program Design Level
- Language Design Level

At a program level subprogram represents a an abstract operation that programmer define rather than primitive operation that are built in to language

Subprogram as an Abstract Operation:

A subprogram definition has two parts:

- Specification
- Implementation

Specification of a program

Encapsulation by Subprogram

Specification of a program:

It includes:

1. The name of the subprogram
2. Signature(also known as proto type) giving the number of arguments ,their order, data type of each as well as number of result their order, and data type of each
3. The action performed by the subprogram

An example of a syntax:

```
float FN(float X, int Y)
```

That specifies the signature as:

FN: real * integer -- real

Subprogram Implementation

A subprogram is implemented using the data structure and operations provided by the programming language. The implementation is defined by the subprogram body , thst consists of local data declaration defining the data structure used by sub program and statements defining the actions to be taken when subprogram is executed. The

declaration and statements are usually encapsulated so that neither the local data nor statements are accessible to user of the subprogram: The user may only invoke the

subprogram with a particular set of arguments

And receives the output results

Subprogram Implementation

The syntax of a subprogram

- Signature of a subprogram
- Declaration of local data objects
- Sequence of statements defining the action of a subprogram

Each invocation of a subprogram requires the arguments of the proper types

Type checking may be same as in data types

- Static Type Checking
- Dynamic Type Checking

SUBPROGRAM DEFINITION & INVOCATION

During the execution of a program if a subprogram is called(invoked) an activation

Of a subprogram is created. When a execution of subprogram is complete the activation is destroyed. If another call is made , a new activation is created .From a single subprogram it is possible to many activation may be created during program execution..A definition is information that present at the time of translation. An activation has a life time-the time during execution between call that creates it and returns that destroys it.

Subprogram Definition & Invocation

To construct a particular activation of a subprogram it is required to split in to parts:

<u>A static part :</u>

Also known as code segment made with the help of constant and executable code. It should be invariant during execution of a subprogram and so a single copy may be shared by all act ovation

<u>A dynamic Part</u>

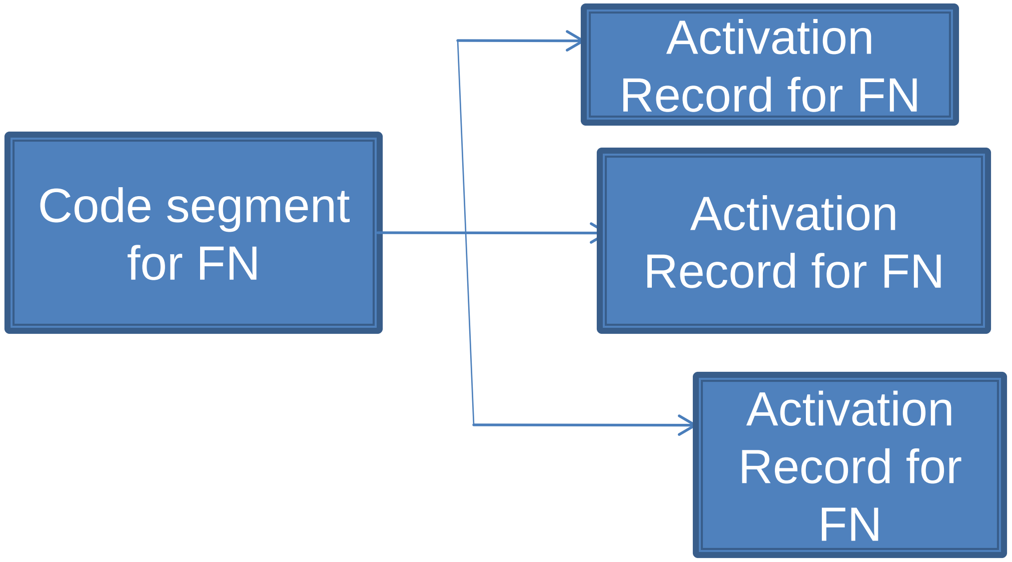
Known as activation record made with the help of parameter, function results and local data some other point like temporary storage areas , return point .

The size and structure of activation record for a subprogram can be find out at translation time

GENERIC SUBPROGRAM

The specification of subprogram lists the number, order and data types of arguments. A generic subprogram is one with a single name but several different definitions defined be different signature.

A generic subprogram is said to be overloaded



Shared Code and Separate Activation Records

Type Definition

Some language provides a flexibility that user can define their own data type. This mechanism is termed as Type Definition

Like in Pascal

```
type Rational =Record
             numerator :integer;
              denominator: integer;
                    end
               var A,B,C: Rational;
```

Syntax is :

```
                Typedef  definition name
 typedef   structRationalType
             {int numerator;
              int denominator;} Rational

               Rational A,B,C; in C langauge
```