Procedures, parameters and sub-programs 13716

Technology, Internet

Procedures, Parameters & Sub-Programs

In any modern programming language, procedures play a vital role in the

construction of any new software. These days, procedures are used instead of

the old constructs of GOTO and GOSUB, which have since become obsolete.

Procedures provide a number of important features for the modern software

engineer:-

Programs are easier to write. Procedures save a large amount of time during

software development as the programmer only needs to code a procedure once, but

can use it a number of times. Procedures are especially useful in recursive

algorithms where the same piece of code has to be executed over and over again.

The use of procedures allows a large and complex program to be broken up into a

number of much smaller parts, each accomplished by a procedure. Procedures also

provide a form of abstraction as all the programmer has to do is know how to

call a procedure and what it does, not how it accomplishes the task.

Programs are easier to read. Procedures help to make programs shorter, and thus

easier to read, by replacing long sequences of statements with one simple

procedure call. By choosing goo procedure names, even the names of the

procedures help to document the program and make it easier to understand.

Programs are easier to modify. When repeated actions are replaced by one

procedure call, it becomes much easier to modify the code at a later stage, and

also correct any errors. By building up the program in a modular fashion via

procedures it becomes much easier to update and replace sections of the program

at a later date, if all the code for the specific section is in a particular

module.

Programs take less time to compile. Replacing a sequence of statements with

once simple procedure call usually reduces the compilation time of the program,

so long as the program contains more than one reference to the procedure!

Object programs require less memory. Procedures reduce the memory consumption

of the program in two ways. Firstly they reduce code duplication as the code

only needs to be stored once, but the procedure can be called many times.

Secondly, procedures allow more efficient storage of data, because storage for a

procedure's variables is allocated when the procedure is called and deallocated

when it returns.

We can divide procedures into two groups:-

Function procedures, are procedures which compute a single value and whose calls

appear in expressions

For example, the procedure ABS is a function procedure, when given a number x,

ABS computes the absolute value of x; a call of ABS appears in an expression,

representing the value that ABS computes.

Proper procedures, are procedures whose calls are statements

For example, the procedure INC is a proper procedure. A call of INC is a

statement; executing INC changes the value stored in a variable.

Procedures have only one real disadvantage: Executing a procedure requires

extra time because of the extra work that must be done both when the procedure

is called, and when it returns.

Most of the time however, the advantages of using procedures heavily outweigh

this minor disadvantage.

Most procedures depend on data that varies from one call to the next, and for

this reason, Modula-2 allows a procedure heading to include a list of

identifiers that represent variables or expressions to supply when calling the

procedure. The programmer can use these identifiers, known as formal

parameters, in the body of the procedure in the same fashion as ordinary

variables.

A call of a procedure with parameters must include a list of actual parameters.

The number of actual parameters must be the same as the number of formal

parameters. Correspondence between the actual and formal parameters is by

position, so the first actual parameter corresponds to the first formal

parameter, the second actual parameter corresponds to the second formal

parameter, and so on. The type of each actual parameter must match the type of

the corresponding formal parameter.

Modula-2 provides to kinds of formal parameters:-

Variable parameters. In a procedure heading, if the reserved word VAR precedes

a formal parameter, then it is a variable parameter. Any changes made to a

variable parameter within the body of the procedure also affect the

corresponding actual parameter in the main body of the program.

Value parameters. If the reserved word VAR does not precede a formal parameter

then it is a value parameter. If a formal parameter is a value parameter, the

corresponding actual parameter is protected from change, no matter what changes are made to the corresponding parameter in the procedure.

To sum up, variable parameters allow information to flow both into and/or out of

a procedure, whereas value parameters are one way and only allow information

into a procedure.

Most Modula-2 systems allow a program to " call" a program module as if it were a

procedure. We call a module used in this way a subprogram module or just a

subprogram. The commands for calling another program are not part of Modula-2

itself, but are provided by a procedure in a library module. The command used

in most Modula-2 systems for calling a sub-program is CALL, and a number of

parameters are usually passed along with this procedure so as to allow the two

programs to communicate with each other, but there is no way to supply

parameters to a subprogram. The parameters passed only indicate things like whether the sub-program was executed correctly and did not terminate early

because of an error.

The primary reason for using subprograms is to reduce the amount of memory

required to execute a program. If a program is too large to fit into memory,

the programmer can often identify one or more modules, that need not exist

simultaneously. The main module can then call these modules as subprograms when

needed. Once a subprogram has completed execution, it returns control to the

main program, which can then call another sub program. All subprograms share

the same area of memory, and because only one is resident at a time, the memory

requirements of the overall program are greatly reduced.