

A Pascal program can consist of modules called units. A unit might consist of some code blocks, which in turn are made up of variables and type declarations, statements, procedures, etc. There are many built-in units in Pascal and Pascal allows programmers to define and write their own units to be used later in various programs.

Using Built-in Units

Both the built-in units and user-defined units are included in a program by the `uses` clause. We have already used the `variants` unit in *Pascal - Variants* tutorial. This tutorial explains creating and including user-defined units. However, let us first see how to include a built-in unit **crt** in your program –

```
program myprog;  
uses crt;
```

The following example illustrates using the **crt** unit –

```
Program Calculate_Area (input, output);  
uses crt;  
var  
    a, b, c, s, area: real;  
begin  
    textbackground(white); (* gives a white background *)  
    clrscr; (*clears the screen *)  
  
    textcolor(green); (* text color is green *)  
    gotoxy(30, 4); (* takes the pointer to the 4th line and 30th column)  
  
    writeln('This program calculates area of a triangle:');  
    writeln('Area =  $\text{area} = \sqrt{s(s-a)(s-b)(s-c)}$ ');  
    writeln('S stands for semi-perimeter');  
    writeln('a, b, c are sides of the triangle');  
    writeln('Press any key when you are ready');  
  
    readkey;  
    clrscr;  
    gotoxy(20, 3);  
  
    write('Enter a: ');  
    readln(a);  
    gotoxy(20, 5);  
  
    write('Enter b:');  
    readln(b);  
    gotoxy(20, 7);  
  
    write('Enter c: ');  
    readln(c);  
  
    s := (a + b + c)/2.0;  
    area := sqrt(s * (s - a)*(s-b)*(s-c));  
    gotoxy(20, 9);  
  
    writeln('Area: ', area:10:3);  
    readkey;  
end.
```

It is the same program we used right at the beginning of the Pascal tutorial, compile and run it to find the effects of the change.

Creating and Using a Pascal Unit

To create a unit, you need to write the modules or subprograms you want to store in it and save it in a file with **.pas** extension. The first line of this file should start with the keyword **unit** followed by the name of the unit. For example –

```
unit calculateArea;
```

Following are three important steps in creating a Pascal unit –

- The name of the file and the name of the unit should be exactly same. So, our unit *calculateArea* will be saved in a file named *calculateArea.pas*.
- The next line should consist of a single keyword **interface**. After this line, you will write the declarations for all the functions and procedures that will come in this unit.
- Right after the function declarations, write the word **implementation**, which is again a keyword. After the line containing the keyword **implementation**, provide definition of all the subprograms.

The following program creates the unit named *calculateArea* –

```
unit CalculateArea;
interface

function RectangleArea( length, width: real): real;
function CircleArea(radius: real) : real;
function TriangleArea( side1, side2, side3: real): real;

implementation

function RectangleArea( length, width: real): real;
begin
    RectangleArea := length * width;
end;

function CircleArea(radius: real) : real;
const
    PI = 3.14159;
begin
    CircleArea := PI * radius * radius;
end;

function TriangleArea( side1, side2, side3: real): real;
var
    s, area: real;
begin
    s := (side1 + side2 + side3)/2.0;
    area := sqrt(s * (s - side1)*(s-side2)*(s-side3));
    TriangleArea := area;
end;

end.
```

Next, let us write a simple program that would use the unit we defined above –

```
program AreaCalculation;
uses CalculateArea, crt;

var
    l, w, r, a, b, c, area: real;

begin
    clrscr;
    l := 5.4;
    w := 4.7;
```

```
area := RectangleArea(1, w);  
writeln('Area of Rectangle 5.4 x 4.7 is: ', area:7:3);  
  
r:= 7.0;  
area:= CircleArea(r);  
writeln('Area of Circle with radius 7.0 is: ', area:7:3);  
  
a := 3.0;  
b:= 4.0;  
c:= 5.0;  
  
area:= TriangleArea(a, b, c);  
writeln('Area of Triangle 3.0 by 4.0 by 5.0 is: ', area:7:3);  
end.
```

When the above code is compiled and executed, it produces the following result –

```
Area of Rectangle 5.4 x 4.7 is: 25.380  
Area of Circle with radius 7.0 is: 153.938  
Area of Triangle 3.0 by 4.0 by 5.0 is: 6.000
```