In this lesson we will take a closer look at this thing called "**eventrecord**".
Heck we are going to take a real close look...so close we will see exactly what the system sees. And we are going to write a small program to help
us look at it. The **eventrecord** is made up of 5 specific items. They are:

1) an event code- this identifies **what** the event was.

2) a **message** code- which provides even more info about that event

3) **when** the event took place

4) data about **where** the mouse was at the time the event took place

5) **modifiers** which help in detailed analysis of the event.

We won't get involved with precisely what all the codes mean within each of these 5 items. It is important now that we understand what the eventrecord is and what info we will be drawing on when we handle these events. Run the following program. experiment by pressing different keys on you keyboard to see the difference in each eventrecord. As indicated mousedown will terminate the program.....but i have found that if you click real fast you will get one or two event records about the mouse. By the way..the where item does not show you the x,y coordinates instead it identifies each pixel with a number...so the upper left hand corner of your screen is x=1,y=1...but where identifies this as 1...x=2,y=1 would be pixel number 2....with me?...ok your screen is approximately 320 pixels by 520 pixels....for an approximate total number of pixels on your screen of just over 166,000. Now, ready for something not known to many macusers? The Event Manager who tracks all this for us... is not limited to the size of your mac screen...(this is sort of obvious since the release of the radius and other full page displays) . The event manager could concieveably handle a screen the size of 4 billion pixels! Thats one heck of a large display!

Ok, on to the program. Here it is.

```
-----------------------------------------------------------------
program eventqueue;                          {program name}

uses  Memtypes,QuickDraw,OSIntf,ToolIntf,FixMath,   {loaded all units}
Graf3D,macprint,packintf,appletalk, speechIntf,
SCSIIntf;
var
whathappened:eventrecord;                    {will tell me
                                              what the system
```

records in the event-
                                          record as far as
                                          keypresses are
                                          concerned}

procedure geteventinfo;   {this is the only
                                          procedure that will
                                           be used in the main
                                          loop/program}

```pascal
begin
if getnextevent(14,whathappened) then
  begin
  writeln('what:',whathappened.what);
  writeln('message:',whathappened.message);
  writeln('when:',whathappened.when);
  writeln('where:',whathappened.where.v,'',whathappened.where.h);
  writeln('modifiers:',whathappened.modifiers);
  writeln;
  end;
end;                                      {end of procedure}

begin                                     {begin main program}
  repeat                                  {repeats the
                                          procedure geteveninfo-
                                          until mousedown}

  geteventinfo;
  until button;
end.                                      {end of program}
```

Don't lose sight of what it is we are really trying to accomplish. Remember we got here
because we wanted to do something more than readln and writeln. So we had to know more
about the event driven programming required by the mac interface...to do that we have to
know how to handle an event...and events are recorded by the event manager as you will
soon see.

Once familiar with **eventrecords** we can go back to handling events as we started in 3rd
grade 1.