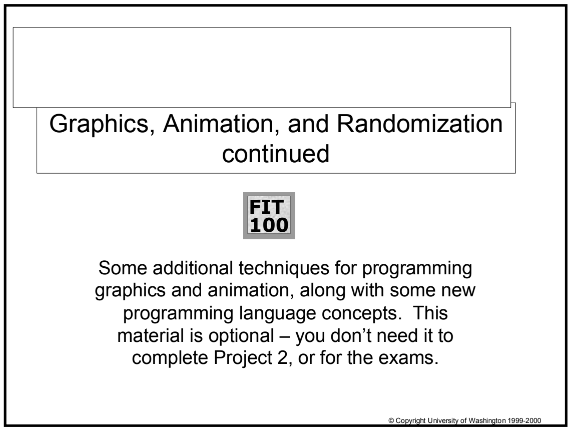
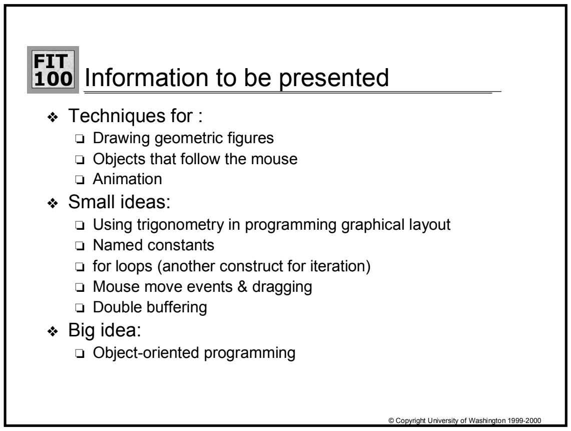


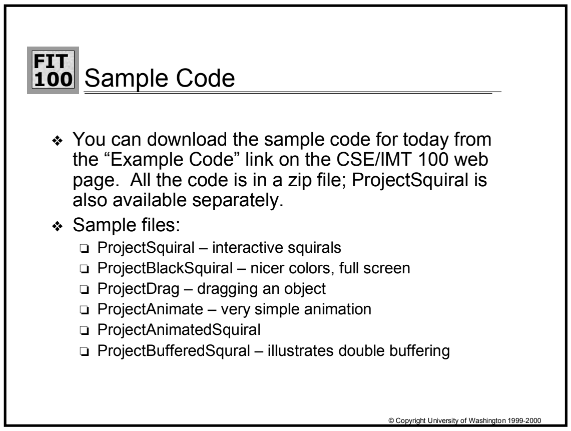
Graphics, Animation, and Randomization continued

Some additional techniques for programming graphics and animation, along with some new programming language concepts. This material is optional – you don't need it to complete Project 2, or for the exams.



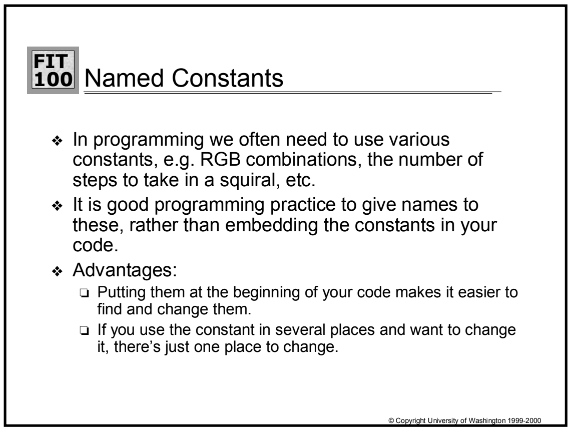
Information to be presented

- Techniques for :
  - Drawing geometric figures
  - Objects that follow the mouse
  - Animation
- Small ideas:
  - Using trigonometry in programming graphical layout
  - Named constants
  - for loops (another construct for iteration)
  - Mouse move events & dragging
  - Double buffering
- Big idea:
  - Object-oriented programming



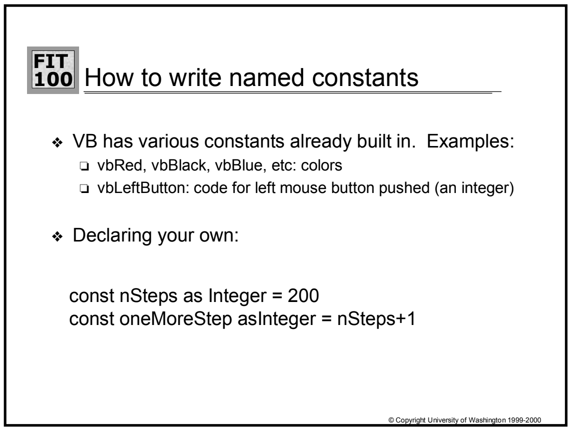
Sample Code

- You can download the sample code for today from the "Example Code" link on the CSE/IMT 100 web page. All the code is in a zip file; ProjectSquiral is also available separately.
- Sample files:
  - ProjectSquiral – interactive squirals
  - ProjectBlackSquiral – nicer colors, full screen
  - ProjectDrag – dragging an object
  - ProjectAnimate – very simple animation
  - ProjectAnimatedSquiral
  - ProjectBufferedSqural – illustrates double buffering



Named Constants

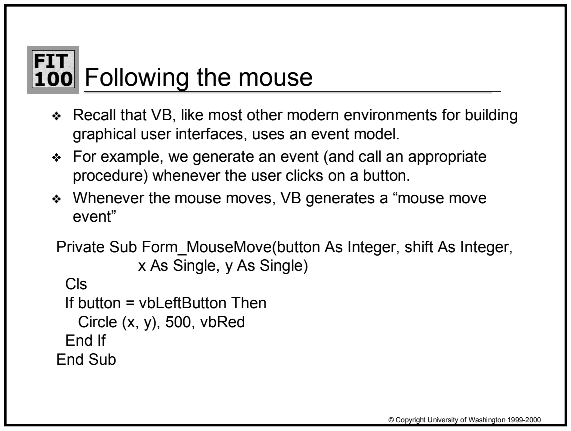
- In programming we often need to use various constants, e.g. RGB combinations, the number of steps to take in a squiral, etc.
- It is good programming practice to give names to these, rather than embedding the constants in your code.
- Advantages:
  - Putting them at the beginning of your code makes it easier to find and change them.
  - If you use the constant in several places and want to change it, there's just one place to change.



How to write named constants

- VB has various constants already built in. Examples:
  - vbRed, vbBlack, vbBlue, etc: colors
  - vbLeftButton: code for left mouse button pushed (an integer)
- Declaring your own:

```
const nSteps as Integer = 200
const oneMoreStep asInteger = nSteps+1
```



Following the mouse

- Recall that VB, like most other modern environments for building graphical user interfaces, uses an event model.
- For example, we generate an event (and call an appropriate procedure) whenever the user clicks on a button.
- Whenever the mouse moves, VB generates a "mouse move event"

```
Private Sub Form_MouseMove(button As Integer, shift As Integer,
              x As Single, y As Single)
  Cls
  If button = vbLeftButton Then
    Circle (x, y), 500, vbRed
  End If
End Sub
```

Animation

- You can do simple animations using the timer control. On each call to the Timer procedure, erase the form and draw the new figure.

```
Private Sub Timer1_Timer()
  Cls
  Circle (x, y), 200, vbBlue
  x = x + 20 * direction
  y = y + 10 * direction
  If x < 100 Or x > 3000 Then
    direction = 0 - direction
  End If
End Sub
```

Double Buffering

- If you are doing a complex animation, it will take VB a bit of time to do the drawing.
- Problem: flicker
- Solution: double buffering
  - Draw the figure in a different picture that is not visible
  - Copy the result into the visible picture