Thinking Through A Program

For a program to compute the desired result it must proceed through a series of logical steps, transforming the inputs into outputs. Figuring out what those logical steps should be is the task of the programmer.
Projects are multipart tasks that span a couple of weeks in which a significant computation is developed.

We will program Sign Find

Enter Your Sign, Please

Aries  Leo  Sagittarius
Taurus  Virgo  Capricorn
Gemini  Libra  Aquarius
Cancer  Scorpio  Pisces

You were born between
July 23 And August 22

Enter the Month and Day of Your Birth

January  July
February  August
March  September
April  October
May  November
June  December

Enter Your Sign, Please

Aries  Leo  Sagittarius
Taurus  Virgo  Capricorn
Gemini  Libra  Aquarius
Cancer  Scorpio  Pisces

You were born between
June 21 And July 22

Were you born after July 6
Sign Finder

- Sign Finder accepts the day and month of a person’s birth and returns the person’s Zodiac sign
- In formulating the logic of the computation, specify the inputs and outputs first
  - Input: A month and a day
  - Output: The name of the Zodiac sign

Since the GUI is the source of the input and the display for the output, it can be designed at this point too.
The radio buttons and the text box of the GUI are the means of presenting input.

The computation takes place when the OK is clicked.

Since clicking a radio button, entering the text box and clicking the OK command button are “events”, the program can be developed by considering what computation is needed in response to each event:

- Month Radio Button -- set up the data for that month
- Day Value Entry -- save the data for later
- OK Command Button -- Determine the sign and print it

How is the sign determined from the month and day?
Consider The Signs

- Notice that a person born in a given month could have one of two signs, depending on the day of birth.

- A July birthday could be either Cancer or Leo.

- Every month is similar: There are two signs possible.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Aries</td>
<td>March 21</td>
<td>April 19</td>
</tr>
<tr>
<td>Taurus</td>
<td>April 20</td>
<td>May 20</td>
</tr>
<tr>
<td>Gemini</td>
<td>May 21</td>
<td>June 20</td>
</tr>
<tr>
<td>Cancer</td>
<td>June 21</td>
<td>July 22</td>
</tr>
<tr>
<td>Leo</td>
<td>July 23</td>
<td>August 22</td>
</tr>
<tr>
<td>Virgo</td>
<td>August 23</td>
<td>September 22</td>
</tr>
<tr>
<td>Libra</td>
<td>September 23</td>
<td>October 22</td>
</tr>
<tr>
<td>Scorpio</td>
<td>October 23</td>
<td>November 21</td>
</tr>
<tr>
<td>Sagittarius</td>
<td>November 22</td>
<td>December 21</td>
</tr>
<tr>
<td>Capricorn</td>
<td>December 22</td>
<td>January 19</td>
</tr>
<tr>
<td>Aquarius</td>
<td>January 20</td>
<td>February 18</td>
</tr>
<tr>
<td>Pisces</td>
<td>February 19</td>
<td>March 20</td>
</tr>
</tbody>
</table>
When a month is chosen, remember the two signs that apply and the day when they change … when the birth day is chosen it is possible to pick the sign

<table>
<thead>
<tr>
<th>Sign</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cancer</td>
<td>June 21</td>
<td>July 22</td>
</tr>
<tr>
<td>Leo</td>
<td>July 23</td>
<td>August 22</td>
</tr>
</tbody>
</table>

Logic for July

loSign = “Cancer”
ihiSign = “Leo”
midpt = 22
On The Specification Of The Day ...

- The day is simply a number that is typed in
- It should be saved in a variable for later use

The text box is a control that will be named “txtDay”. Its content is referred to as its property “Text”. To refer to any property write `<control name>..<property>`

Logic for Textbox

...  
`dayPick = txtDay.Text`  
...
On Clicking On OK

- With the month chosen and the day chosen, it is possible to figure out the sign
  - If the day is on the \textit{midpt} or before, it's the earlier sign
  - If the day is after the \textit{midpt}, it's the later sign

\begin{verbatim}
... If dayPick <= midpt Then
    lblSign.Caption = loSign
Else
    lblSign.Caption = hiSign
End If
lblSign.Visible = True
lblYour.Visible = True
...
\end{verbatim}

This label control will be called "\texttt{lblSign}"
Having Brained Out The Logic ...

- The following steps achieve the result
- Create the GUI
- Declare the four variables
  - loSign, a string
  - hiSign, a string
  - midpt, an integer
  - dayPick, an integer
- Set loSign, hiSign and midpt in radio buttons
- Set dayPick to the text input
- For the OK click event, incorporate the If-statement and set the visibility of the two labels