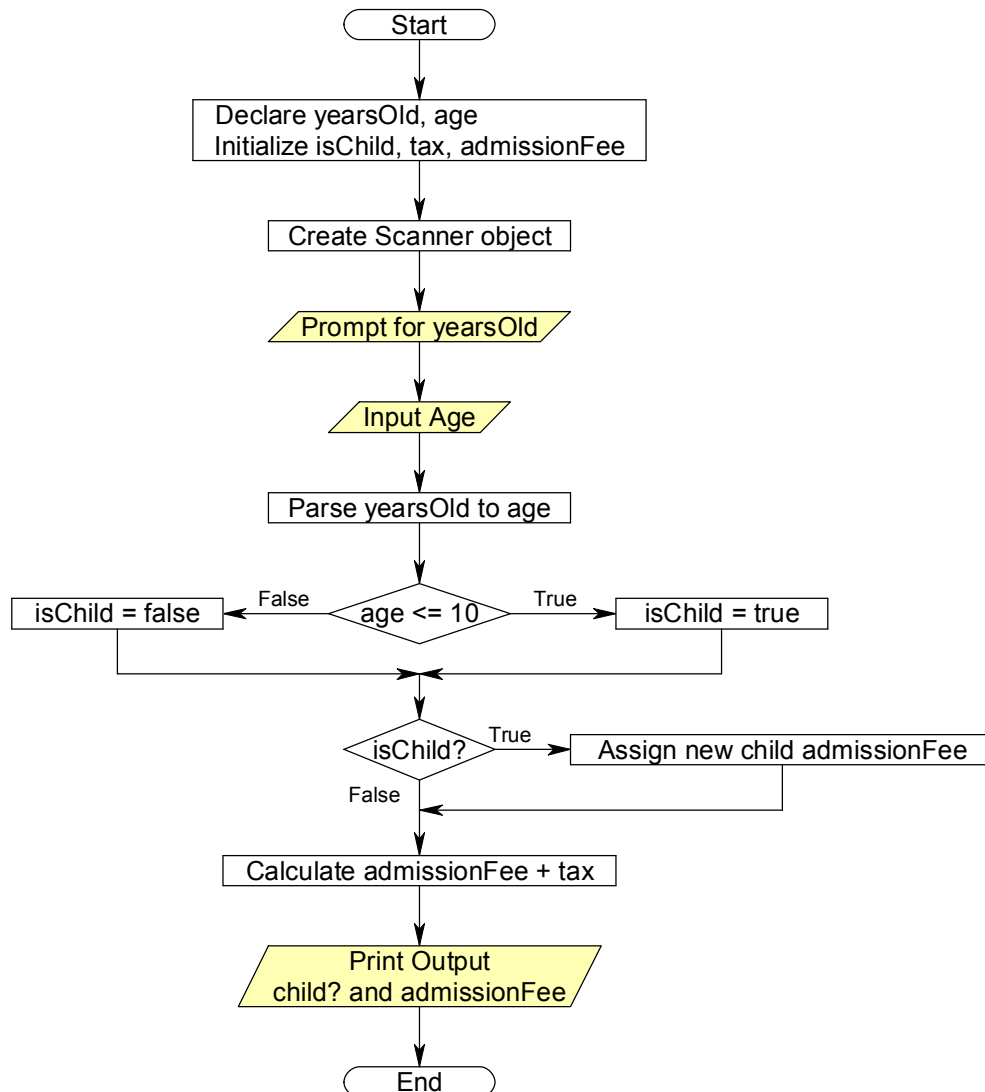


## 4.03 Virtual Lecture Notes (Part 1)

The AdmissionFee\_v1 class is a program designed to calculate the admission fee to an event based on a person's age. Carefully study the structure of this flowchart; find the input, processing, and output sections.



- Open the AdmissionsFee\_v1 class and compare the source code to the flowchart.
- Analyze the program line-by-line and make sure you understand the syntax and purpose of each statement in the program.
- Run the program and observe the performance and the output.

The segment of code shown below is where “the **boolean** meets the code” in the program.

```
...  
< 8>      boolean isChild;  
< 9>      double admissionFee = 6.00;  
...  
<17>      isChild = age <= 10;  
<18>  
<19>      if(isChild)  
<20>          admissionFee = 4.50;  
<21>  
<22>      admissionFee += admissionFee * tax;  
...
```

Line < 8> declares **isChild** to be a **boolean** primitive data type.

Line < 9> declares **admissionFee** to be a **double** primitive data type and assigns 6.00 as the variable’s initial value.

Line <17> is a **boolean** expression that evaluates whether the age entered by the user is less than or equal to 10. If the age is less than or equal to 10, **true** is assigned to the **isChild** variable; otherwise **false** is assigned.

Line <19> evaluates the value of **isChild** (i.e. **true** or **false**).

Line <20> assigns a new value to the **admissionFee** variable (4.50); otherwise the default value assigned in Line <8> is retained (6.00).

Line <22> calculates the admission fee using the value appropriate for a child or someone who is “not a child”.

Conditional statements must be syntactically correct. Improper **boolean** expressions, missing parentheses, and misplaced semicolons are frequent sources of error. Stylistically, readability will be improved by following a strict indentation scheme. Using **boolean** variables is good programming practice and it forces you to be aware of the logical design of algorithms.