

## Numerical Program Input

- The cin function with the extraction operator (>>) receives text from keyboard.
- This text converted automatically to the data type of variable that it is extracted to
  - Ignores leading whitespace characters
  - Stops at first inappropriate character
- Examples:
 

```
cin >> nScore;
cin >> fNum1 >> fNum2 >> fNum3;
```

Copyright © 2001 R.M. Laurie 1

## Character Program Input

- Reads first non-whitespace character
  - cin >> cChar;
  - cin skips leading whitespace characters
- Reads first character including whitespace
  - cin.get(cChar);
- Example
 

```
Input data:  gk
cin >> cChar;
cin.get(cChar);
```

What does cChar = ?

Copyright © 2001 R.M. Laurie 2

## Ignoring Characters from Input

- cin.ignore is used to ignore a specified number of characters from the istream.

```
cin.ignore(#, '\n');
```

# = ignore the next # characters  
 '\n' = or skip until newline character is read whichever comes first

Copyright © 2001 R.M. Laurie 3

## Input Example

- Example:
 

```
Input data:  gk  ab␣xyz ␣

#include <iostream>
using namespace std;
int main()
{
    char cA, cB, cC, cD;

    cin >> cA;
    cin.ignore(4, '\n');
    cin >> cB;
    cin >> cC;
    cin.get(cD);
    cout << cA << cB << cC << cD;
    return 0;
}
```

Copyright © 2001 R.M. Laurie 4

### Creating Fault Tolerant Input

---

- For Numerical Entries  
`cin >> fTemperature;`  
`cin.ignore(100, '\n');`
- For Character Entries  
`cin >> cQuestion;`  
`cin.ignore(100, '\n');`

```
cin.get(cQuestion);
cin.ignore(100, '\n');
```

Copyright © 2001 R.M. Laurie | 5

### Program Prompt and Receive

---

- To make your programs user friendly it is best to prompt for data entry by displaying a message.
- Receive a value from keyboard.
- Then perform error detection on entered data.
- Finally display entered data with results.
- This requires use of selection and loop structures (future)

```

graph TD
    Prompt[Prompt] --> Receive[Receive]
    Receive --> Error{Entry Error?}
    Error -- YES --> Prompt
    Error -- NO --> Print[Print Results]
    
```

Copyright © 2001 R.M. Laurie | 6

### Flow of Control

---

- Flow of control
  - Definition: The sequence in which the computer executes statements of the program.
- Sequential Control Structure
- Selection (Branching) Control Structure
  - Relational and Logical Operators
- Repetition (Loop) Control Structure

Copyright © 2001 R.M. Laurie | 7

### Sequential Control Structure

---

- The Sequential Control Structure is simply the execution of program instructions from the top to bottom.
- It is characterized by a flow chart construct without branches.

```

graph TD
    Start([Start]) --> S1[Statement 1;]
    S1 --> S2[Statement 2;]
    S2 --> S3[Statement 3;]
    
```

Copyright © 2001 R.M. Laurie | 8

## Selection Control Structure

- The Selection Control Structure
  - Branching Control Structure
  - Conditional Control Structure
- Decision making control
- Tests an Assertion Statement
  - Evaluated as True or False (Humans)
  - Evaluated as Yes or No (Humans)
  - Evaluated as 1 or 0 (Computers)

Copyright © 2001 R.M. Laurie | 9

## Relational Operators

- Relational operators are used to compare two data objects.
- The result of the comparison is either TRUE (1) or False (0).
 

|             |                     |
|-------------|---------------------|
| == Equal to | != Not Equal to     |
| > Greater   | >= Greater or Equal |
| < Less      | <= Less or Equal    |
- Note the difference between == and = operator

Copyright © 2001 R.M. Laurie | 10

## Operators Precedence

```

( )
float() int() sizeof() !
* / %
+ -
<< >>
< <= > >=
== !=
=
    
```

Copyright © 2001 R.M. Laurie | 11

## if Selection Control Structure

- Characterized by a diamond shaped flow chart construct, containing an assertions with two possible outcomes branches (True or False).

```

graph TD
    Start(( )) --> Receive[Receive Score]
    Receive --> Decision{Score >= 90}
    Decision -- True --> Print[Print Grade= A]
    Print --> Join(( ))
    Decision -- False --> Join
    Join --> Start
    
```

Copyright © 2001 R.M. Laurie | 12

### if Selection Control Structure

- Characterized by a diamond shaped flow chart construct, containing an assertions with two possible outcomes branches (True or False).
- Code Inserted

```

graph TD
    Start(( )) --> Input[cin >> nScore;]
    Input --> Decision{if(nScore >= 90)}
    Decision -- True --> TrueBlock[cout << "Grade = A";]
    Decision -- False --> Join(( ))
    TrueBlock --> Join
    Join --> Exit(( ))
    
```

Copyright © 2001 R.M. Laurie 13

### if Selection Control Structure (Single statement syntax)

```

cin >> nScore;
if(nScore < 60)
    cout << "Grade = F";

//Next Statement
    
```

Copyright © 2001 R.M. Laurie 14

### *if* Compound Selection Control Structure

```

graph TD
    Start(( )) --> Input[cin >> nScore;]
    Input --> Decision{if(nScore < 60)}
    Decision -- True --> TrueBlock[cout << "Grade = F";  
Statement 2;  
Statement 3;]
    Decision -- False --> Join(( ))
    TrueBlock --> Join
    Join --> Statement4[Statement 4;]
    Statement4 --> Exit(( ))
    
```

Copyright © 2001 R.M. Laurie 15

### if Selection Control Structure (Compound statement syntax)

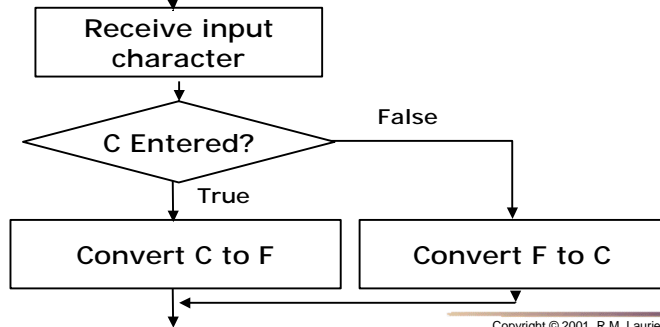
```

cin >> nScore;
if(nScore < 60)
{
    cout << "Grade = F";
    Statement 2;
    Statement 3;
}
Statement 4;
    
```

Copyright © 2001 R.M. Laurie 16

## if - else Selection Structure

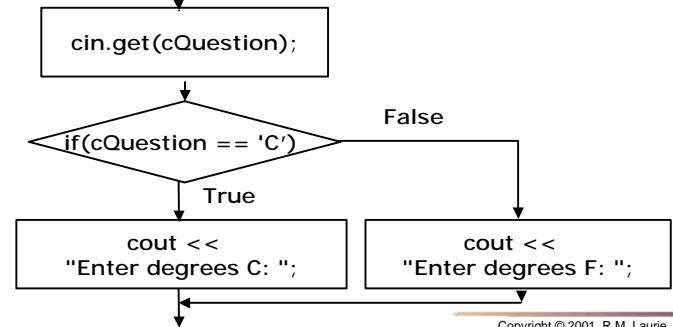
- Characterized by a diamond shaped flow chart construct, containing an assertions with two possible outcomes branches (True or False).



Copyright © 2001 R.M. Laurie 17

## if - else Selection Structure

- Characterized by a diamond shaped flow chart construct, containing an assertions with two possible outcomes branches (True or False).



Copyright © 2001 R.M. Laurie 18

## if - else Selection Structure (Single statement syntax)

```

cin.get(cQuestion);
if(cQuestion == 'C')
    cout << "Enter degrees C: ";
else
    cout << "Enter degrees F: ";
  
```

Copyright © 2001 R.M. Laurie 19

## if - else Selection Structure (Compound statement syntax)

```

cin.get(cQuestion);
if(cQuestion == 'C')
{
    cout << "Enter degrees C: ";
    cin >> fTemp;
    fTemp = ((fTemp*180) / 100)+32;
}
else
{
    cout << "Enter degrees F: ";
    More Statements;
}
  
```

Copyright © 2001 R.M. Laurie 20