Flow of Control

- Definition: The sequential execution of statements in a program
  - Sequential Control Structure (Top-Bottom)
    - It is characterized by a flow chart construct without branches.
  - Selection Control Structure (Branching)
    - 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)

 1

Relational Operators

- Relational operators are used to compare two data objects.
- The result of the comparison is either true or false.

==	Equal to	!=	Not Equal to
>	Greater	>=	Greater or Equal
<	Less	<=	Less or Equal

- Note the difference between == and = operator

 2

Arithmetic Operators Precedence

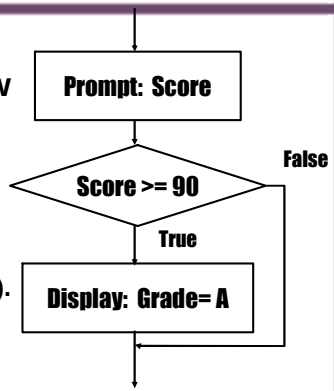
(Highest to Lowest)

.	Property access of an object
( )	Defines order of operation
- ++ --	Minus, Increment, Decrement
!	Logical NOT Operator
* / %	Multiply, Division, Remainder
+ -	Addition, Subtraction
< <= > >=	} Relational Operators
== !=	
&&	Logical AND Operator
||	Logical OR Operator
= += -= *= /= %=	Compound Assignment

 3

if Selection Control Structure

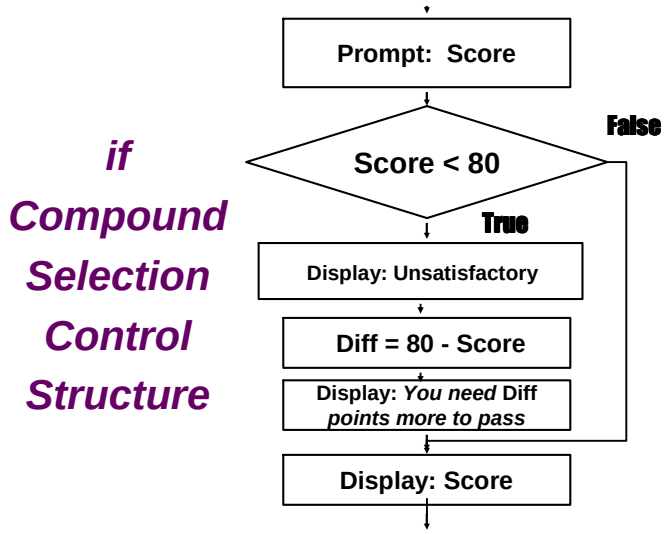
- Characterized by a diamond shaped flow chart construct, containing an assertions with two possible outcomes branches (True or False).




```
if(Score >= 90)
    System.out.println("Grade = A");
```

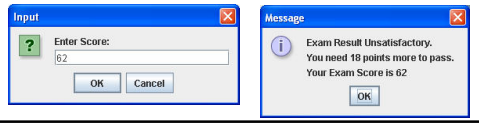
 4

if Compound Selection Control Structure



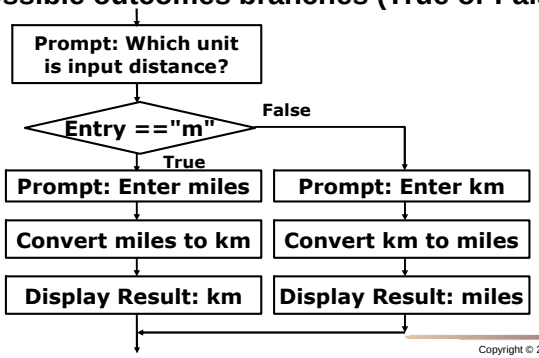

```
import javax.swing.*;
public class GradePF
{
  public static void main(String args[])
  {
    int nScore, nDiff;
    String sEntry, sOutput = "";

    sEntry = JOptionPane.showInputDialog(null,"Enter Score:");
    nScore = Integer.parseInt(sEntry);
    if(nScore < 80)
    {
       sOutput = "Exam Result Unsatisfactory.";
       nDiff = 80 - nScore;
       sOutput+="\nYou need "+nDiff+" points more to pass.\n";
    }
    sOutput += "Your Exam Score is " + nScore;
    JOptionPane.showMessageDialog(null,sOutput);
    System.exit(0);
  }
}
```

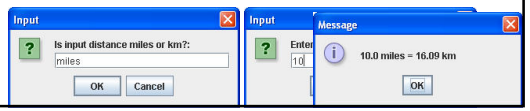


if - else Selection Structure

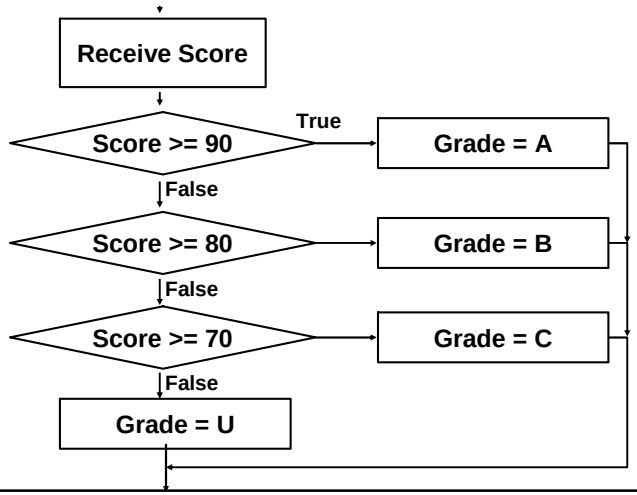
- Characterized by a diamond shaped flow chart construct, containing an assertions with two possible outcomes branches (True or False).



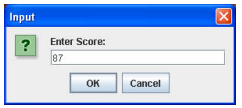

```
import javax.swing.*;
public class MileKm
{
  public static void main(String args[])
  {
    double dMile, dKm;
    String sEntry;
    char cSelect;
    sEntry = JOptionPane.showInputDialog(null,
                "Is input distance miles or km?:");
    cSelect = sEntry.charAt(0);
    if(cSelect == 'm')
    {
      sEntry = JOptionPane.showInputDialog(null,"Enter miles:");
      dMile = Double.parseDouble(sEntry);
      dKm = dMile * 1.609;
      JOptionPane.showMessageDialog(null, dMile+" miles = "+dKm+" km");
    }
    else
    {
      sEntry = JOptionPane.showInputDialog(null,"Enter kilometers:");
      dKm = Double.parseDouble(sEntry);
      dMile = dKm / 1.609;
      JOptionPane.showMessageDialog(null, dKm+" km = "+dMile+" miles");
    }
    System.exit(0);
  }
}
```

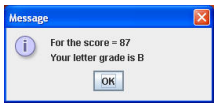


if - else if - else Structure




```
import javax.swing.*;
public class Grade
{
  public static void main(String args[])
  {
    int nScore;
    String sEntry, sOutput;
    char cGrade;
    sEntry = JOptionPane.showInputDialog(null,"Enter Score:");
    nScore = Integer.parseInt(sEntry);
    if(nScore >= 90)
      cGrade = 'A';
    else if(nScore >= 80)
      cGrade = 'B';
    else if(nScore >= 70)
      cGrade = 'C';
    else
      cGrade = 'U';
    sOutput = "For the score = " + nScore
     + "\nYour letter grade is " + cGrade;
    JOptionPane.showMessageDialog(null, sOutput);
    System.exit(0);
  }
}
```





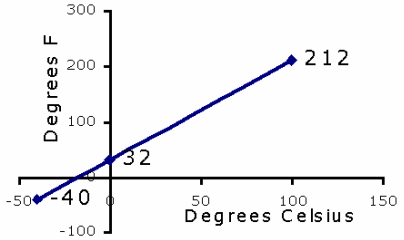
Problem Solving Phase

- Write Program Specifications
  - Analysis of requirements
  - Program specifications description
    - Describe what the goals of the program
    - Describe appearance of input and output
- Algorithm Design
  - Mathematical Analysis and Algorithm
  - Flow Chart to describe event sequencing
- Verify algorithm
  - Test with known data
  - Solve manually

 11

Algorithm Design - Mathematical

- Mathematical Description
  - Boiling point
    F =212
    C = 100
  - Freezing point
    F = 32
    C = 0



Y = MX + B

F = (180 / 100) C + 32
= (9/5) C + 32
= 1.8 C + 32

 12

Verify Algorithm

- Testing with known data
  - Boiling point
    F =212 C = 100
  - Freezing point
    F = 32 C = 0
  - Collect Data
    - Bank thermometer
    - Radio weather report
- Solve manually by hand using calculator

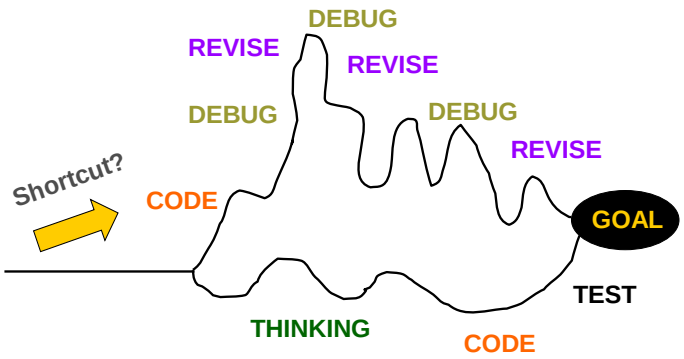
 13

Implementation Phase

- Translate Algorithm into Code
  - Create source code file with syntax of JavaScript language and HTML
  - Run to detect *syntax errors*
- Test Program
  - Test with known data
  - Detects program *logic errors*
  - Often requires several iterations
  - May require re-evaluation of specifications and algorithms

 14

Coding First Is No Shortcut?



 15

Conditional Exercises

- Create program that converts temperatures between Fahrenheit and Celsius
  - Prompt for which Conversion
  - Prompt for the temperature
  - Print code and browser display
- Create an employee's pay program
  - Prompt for name, pay rate, and hours
  - Overtime rate is 1.5x normal pay rate
  - Subtract 15% withholding tax
  - Calculate pay check amount

 16