SELECTION STATEMENTS (2)

Nested if
Switch

CSC 111

Outline

1. Nested if
2. Switch
3. Programming hint

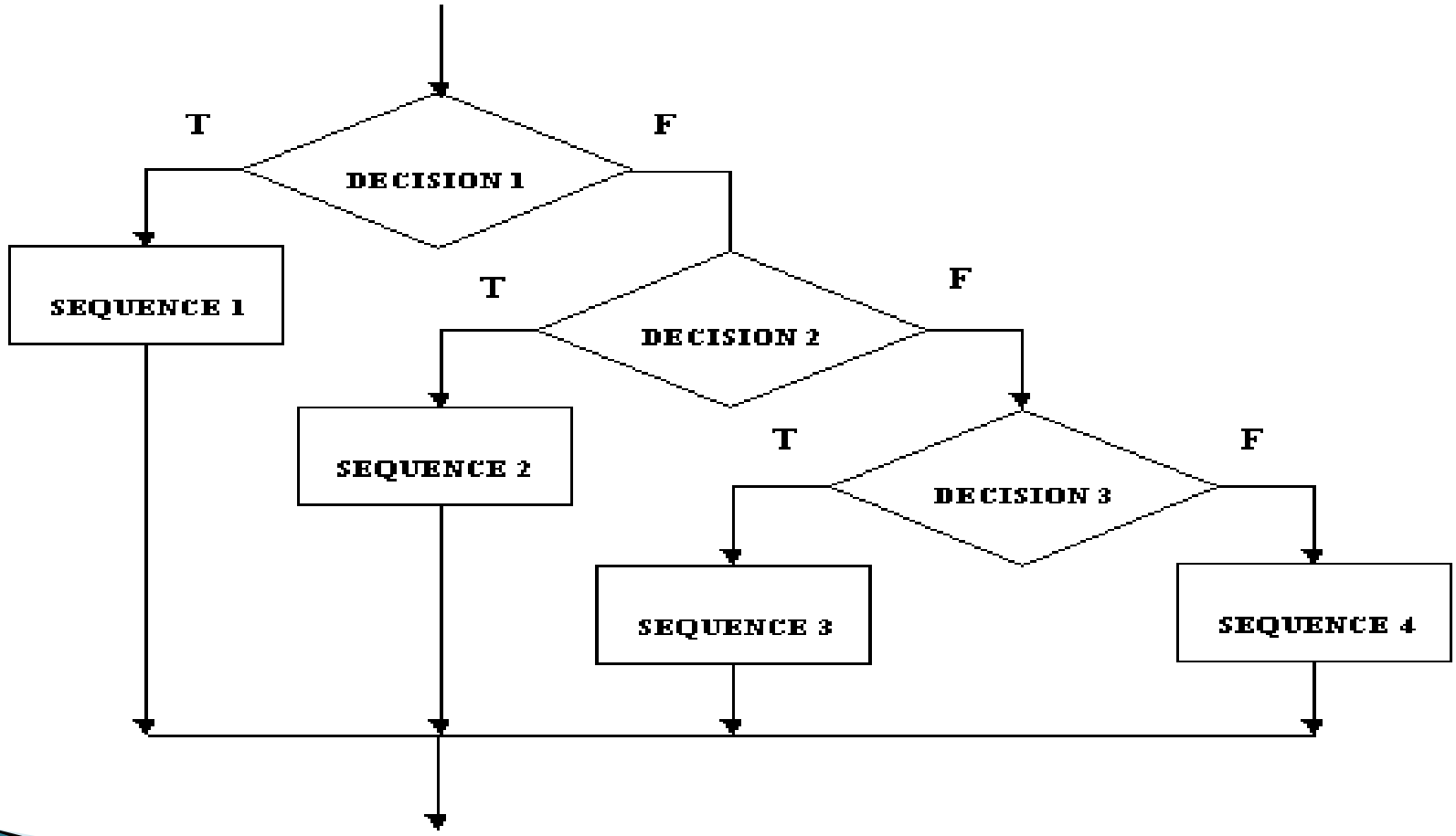
1.Multiple Selection: Nested if

- Syntax:

```
if (expression1)
     statement1
else
    if
(expression2)
        statement2
    else
        statement3
```

- Multiple *if* statements can be used if there is more than two alternatives
- *else* is associated with the most recent *if* that does not have an else.

1. Nested if


T
F
DECISION 1
SEQUENCE 1
T
F
DECISION 2
SEQUENCE 2
T
F
DECISION 3
SEQUENCE 3
SEQUENCE 4

1. Nested if

PROGRAM 1 – ANALYSIS

Write a program that identifies if a number is positive, negative or zero.

INPUT	Number (variable: number, type: double or int)
OUTPUT	An appropriate message (variable: message, type: string)
PROCESS	

```
if (number < 0)
   message = "The number is negative"
otherwise
   if (number > 0)
      message = "The number is positive"
    otherwise
      message = "The number is zero"
```

1. Nested if

PROGRAM 1 – ALGORITHM

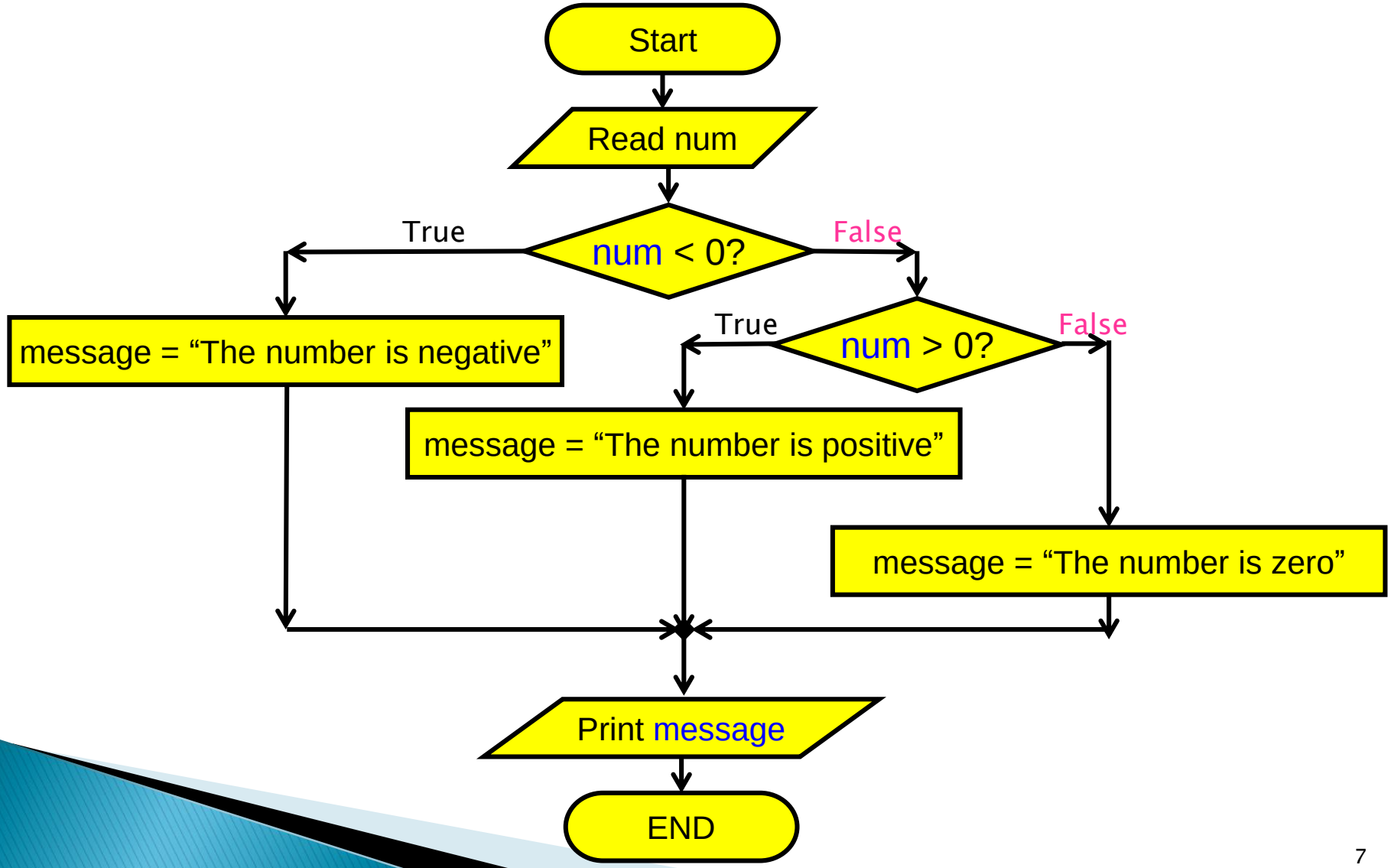
1. Start the program

2. Read the number and save it in the variable num
3. if (num < 0)
    message = "The number is negative"
  else
    if (num > 0)
      message = "The number is positive"
    else
      message = "The number is zero"
4. Print message

5. End the program

1. Nested if

PROGRAM 1 – FLOWCHART



1. Nested if

PROGRAM 1 – CODE

```
// import necessary libraries
import java.util.*;                    //contains the class Scanner
public class nestedIf1
{
   static Scanner console = new Scanner (System.in);
   public static void main (String[] args)
     {
          // Declaration section: to declare needed variables
             int num;
             String message;
          // Input section: to enter values of used variables
             System.out.println ("Enter an integer number");   //prompt
             num = console.nextInt();
          // Processing section: processing statements
             if (num < 0)
                message = "The number is negative";
             else
                 if (num > 0)
                    message = "The number is positive";
                 else
                      message = "The number is zero";
          // Output section: display program output
             System.out.printf ("%25s", message);
     } // end main
} // end class
```

1. Nested if

PROGRAM 1 – NOTES

```
// Processing section: processing statements
    if (num < 0)
        message = "The number is negative";
    else
        if (num > 0)
            message = "The number is positive";
        else
            message = "The number is zero";
```

Each else corresponds to the closer if

1. Nested if

PROGRAM 1 – NOTES

```
// Processing section: processing statements
        if (num < 0)
          message = "The number is negative";
        else
            {
              message = "The number is not negative";
              if (num > 0)
                message = "The number is positive";
              else
                message = "The number is zero";
            } //end else... if (num < 0)
```

Block statements may be used as shown above.

1. Nested if

PROGRAM 2 – ANALYSIS

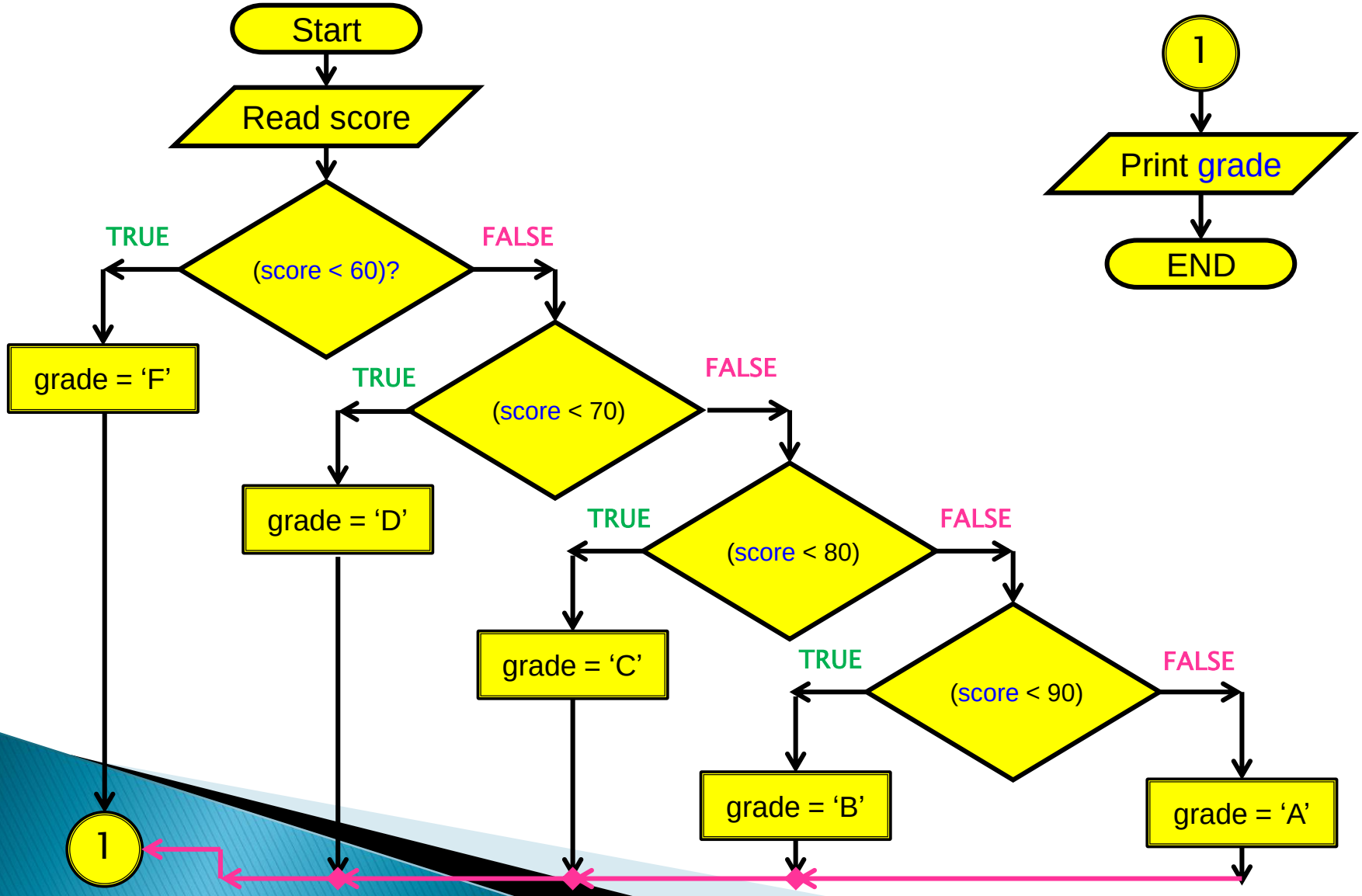
Write a program that calculates the letter grade of a student.

INPUT	Student's score (variable: score, type: double)
OUTPUT	Student's letter grade (variable: grade, type: char)
PROCESS	

```
if (score less than 60.0) grade = 'F'
otherwise
    if (score between 60.0 and 70.0) grade = 'D'
    otherwise
        if (score between 70.0 and 80.0) grade = 'C'
        otherwise
            if (score between 80.0 and 90.0) grade = 'B'
            otherwise grade = 'A'
```

1. Nested if

PROGRAM 2 – FLOWCHART



1. Nested if

PROGRAM 2 – CODE

```
// import necessary libraries
import java.util.*;                    //contains the class Scanner
public class nestedIf2
{
   static Scanner console = new Scanner (System.in);
   public static void main (String[] args)
     {
          // Declaration section: to declare needed variables
             double score;
             char grade;
          // Input section: to enter values of used variables
             System.out.println ("Enter score");   //prompt
             score = console.nextDouble();
          // Processing section: processing statements
```

1. Nested if

PROGRAM 2 – CODE (cont'd)

```
// Processing Section: processing statements
if (score < 60.0)
   grade = 'F';
else
   if (score < 70.0)
      grade = 'D';
   else
      if (score < 80.0)
         grade = 'C';
      else
         if (score < 90.0)
            grade = 'B';
         else
            grade = 'A';
// Output Section: display program output
System.out.printf ("Student's grade = %c", grade);
  } //end main
} //end class
```

1. Nested if

- The following code does not work as intended. Trace for score = 62.

```
if (score > 60.0)
   if (score > 65.0)
      System.out.printf ("Score is greater than 65");
 else
   System.out.printf ("Score is less than 60");
```

```
Enter student's score
62.0
Score is less than 60
```

- This is corrected as follows:

```
if (score > 60.0)
{
   if (score > 65.0)
      System.out.println ("Score is greater than 65");
}
else
   System.out.println ("Score is less than 60");
```

```
Enter student's score
62.0
```

IMPORTANT

1. The switch Statement

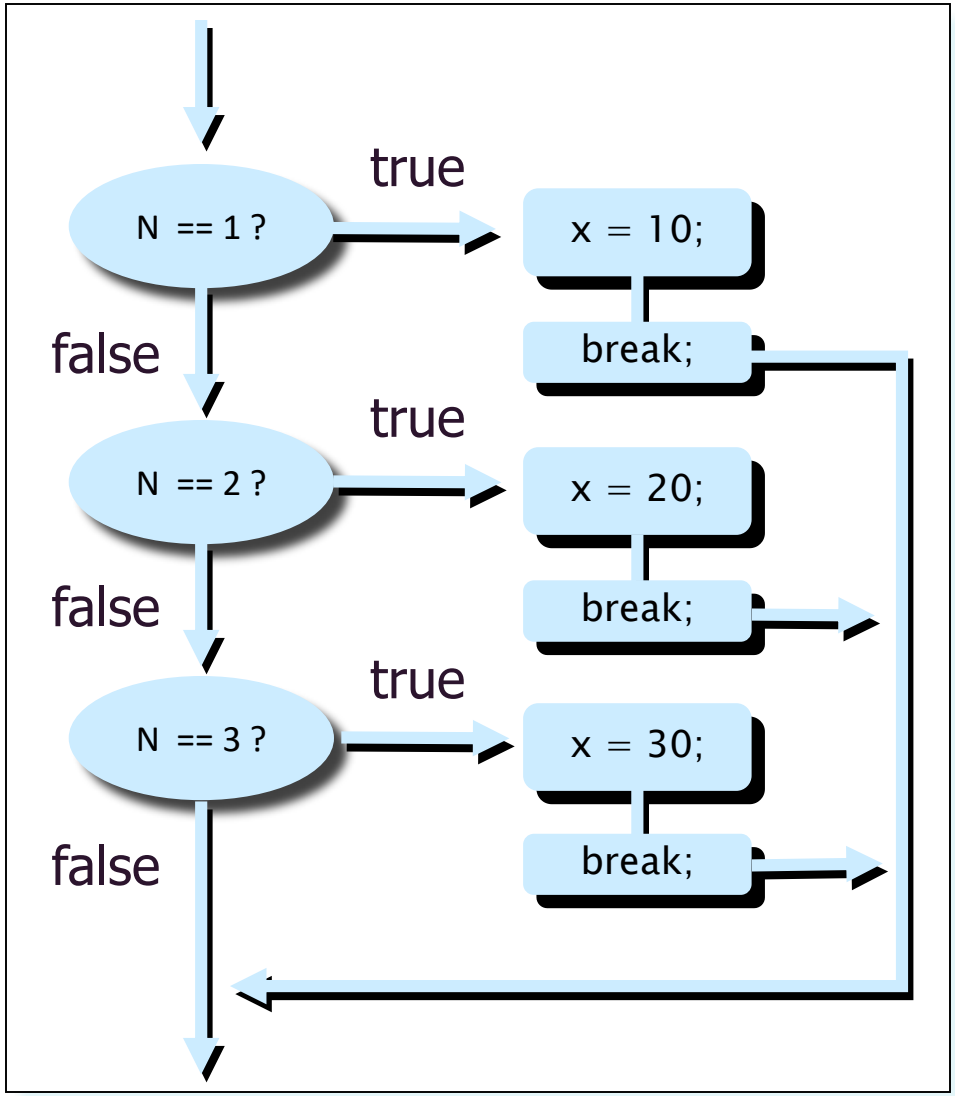
SYNTAX

```
switch (identifier)
  {
    case value 1:
                    statement(s) 1;
                    break;
    case value 2:
                    statement(s) 2;
                    break;
    ---
    ---
    ---
    case value n:
                    statement(s) n;
                    break;
    default:
                    statement(s);
   } //end switch
```

- Expression is also known as selector.
- Expression can be an identifier.
- Value can't be double or float.

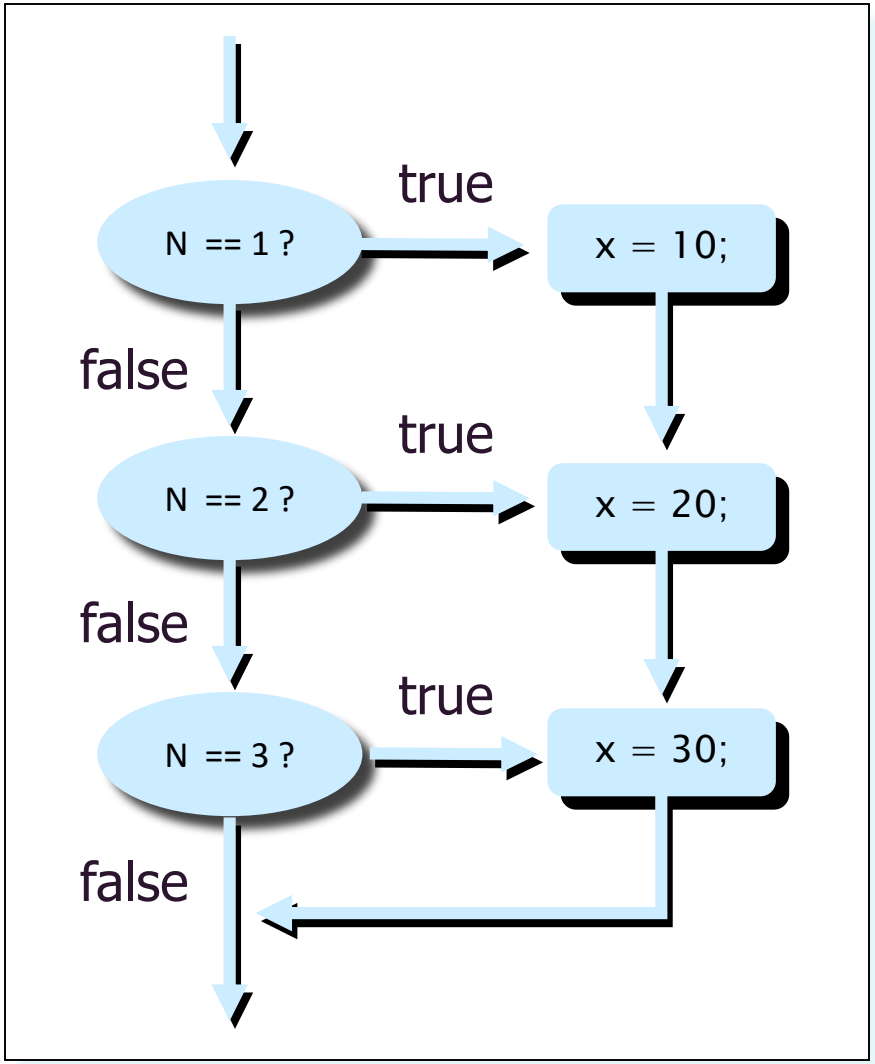
switch With Break Statement

```
switch (N) {
    case 1: x = 10;
            break;
    case 2: x = 20;
            break;
    case 3: x = 30;
            break;
}
```



switch With NO Break Statement

```
switch (N) {
    case 1: x = 10;
    case 2: x = 20;
    case 3: x = 30;
}
```



Switch With Break And Default Statements

```
char grade=read.next().charAt(0);
switch (grade)
{
   case 'A': System.out.println("The grade is A.");
          break;
   case 'B': System.out.println("The grade is B.");
          break;
   case 'C': System.out.println("The grade is C.");
          break;
   case 'D': System.out.println("The grade is D.");
          break;
   case 'F': System.out.println("The grade is F.");
          break;
   default:  System.out.println("The grade is invalid.");
}
```

New enhanced switch syntax

```
char grade=read.next().charAt(0);
switch (grade)
{
   case 'A'-> System.out.println("The grade is A.");
case 'B'-> System.out.println("The grade is B.");
case 'C'-> System.out.println("The grade is C.");
   case 'D'-> System.out.println("The grade is D.");
case 'F'-> System.out.println("The grade is F.");
   default -> System.out.println("The grade is invalid.");
}
```

Same example with Nestedif

```
if (grade == 'A')
      System.out.println("The grade is A.");
else
    if (grade == 'B')
        System.out.println("The grade is B.");
else
    if (grade == 'C')
        System.out.println("The grade is C.");
else
    if (grade == 'D')
        System.out.println("The grade is D.");
else
    if (grade == 'F')
        System.out.println("The grade is F.");
else
      System.out.println("The grade is invalid.");
```

1.3 Programming hint

EXAMPLE 3

- When the same action is to be taken for several case labels, then we may write the program as follows:

```
public static void main (String[] args)
{
   //Declaration section
   Scanner read = new Scanner (System.in);
   char letter;
   String vowel = "This is a vowel";
   //input section
   letter = read.next().charAt(0);
   //processing section
   switch (letter)
   {
      case 'a':
      case 'e':
      case 'o':
      case 'i':
      case 'u': System.out.println (vowel);
                break;
     default: System.out.println ("This is not a vowel");
    } //end switch
} //end main
```

This is equivalent to:

```
if (letter=='a') || (letter=='e') ||
(letter=='o') || (letter=='i') ||
(letter=='u')
      System.out.println (vowel);
```

1.3 New enhanced switch syntax

EXAMPLE 3

- Using the enhanced version, It could be rewritten as following

```
public static void main (String[] args)
{
   //Declaration section
   Scanner read = new Scanner (System.in);
   char letter;
   String vowel = “This is a vowel”;
   //input section
   letter = read.next().charAt(0);
   //processing section
 switch (letter) {
 case 'a', 'e', 'i', 'o', 'u' -> System.out.println(vowel);
default -> System.out.println("This is not a vowel");
  //end switchtch
} //end main
```

2. switch vs. nested if

- The switch statement is an elegant way to apply multiple selections. It is less confusing.
- However, the programmer is forced sometimes to use the nested if. Examples of such cases include:
  - If the selection involves a range of values.
    - Example: if (score >= 60) && (score < 70)
  - If the selector's type is double or float.
    - Example: if (salary == 5000.0)

Self-Check Exercises (1)

- Given that speed = 75 and fee = 0 what is the output of the following code

```
if (speed > 35)
fee = 20.0;
else if (speed > 50)
fee = 40.0;
else if (speed > 75)
fee = 60.0;
System.out.printf ("Fee = %6f.1", fee);
```

```
if (speed < 35)
fee = 20.0;
if (speed < 50)
fee = 40.0;
else if (speed > 75)
fee = 60.0;
System.out.printf ("Fee = %6f.1", fee);
```

```
if (speed < 35)
{
if (speed < 50)
fee = 40.0;
}
else fee = 60.0;
System.out.printf ("Fee = %6f.1", fee);
```

Self-Check Exercises (2)

- Given that salary = 2000.0, what is the output of the following:

```
double lowtax = 0.15;
double hightax = 0.25;
double tax, salary;
tax = (salary >=5000.0) ? lowtax : hightax;
System.out.println (tax);
```

- Given that transactions = 500, what is the output of the following:

```
boolean vip;
int transactions, offer = 10;
vip = (transactions >= 200) ? (offer > 5) : (offer < >=5);
System.out.println (vip);
```

Self-Check Exercises (3)

Write a complete program that calculates and prints the bill for Riyadh's power consumption. The rates vary depending on whether the user is residential, commercial, or industrial. A code of R corresponds to a Residential, C corresponds to a Commercial, and I to Industrial. Any other code should be treated as an error.

The program should read the power consumption rate in KWH (Kilowatt per Hour); then it calculates the due amount according to the following:

The rate is SAR 5 per KWH for Residential, SAR 10 per KWH for Commercial and SAR 20 per KWH for Industrial.

Self-Check Exercises (1)

- Solve the following program using the switch statement:

Write a complete program that calculates and prints the bill for Riyadh's power consumption. The rates vary depending on whether the user is residential, commercial, or industrial. A code of R corresponds to a Residential, C corresponds to a Commercial, and I to Industrial. Any other code should be treated as an error.

The program should read the power consumption rate in KWH (Kilowatt per Hour); then it calculates the due amount according to the following:

The rate is SAR 5 per KWH for Residential, SAR 10 per KWH for Commercial and SAR 20 per KWH for Industrial.

Self-Check Exercises (2)

Write a complete program that calculates and prints the bill for a cellular telephone company. The company offers two types of service: regular and premium. The regular service is coded as 'r' or 'R'; the premium service is coded as 'p' or 'P'. Any other character should be treated as an error.

Rates vary based on the type of service and computed as follows:

Regular service: $10.00 plus first 50 minutes are given free. Charges for over 50 minutes are $0.20 per minute.

Premium service: $25.00 plus:

- For calls made from 6:00 am and 6:00 pm, the first 75 minutes are free; charges for over 75 minutes are $0.10 per minute.
- For calls made from 6.00 pm and 6:00 am, the first 100 minutes are free; charges for over 100 minutes are $0.05 per minute.