



Switch Statement (Outsource: 4-27 – 4-31)

Often a complicated **if-else** statement can be simplified by rewriting it as a switch statement. The only stipulation is that the data being compared must be **ordinal** in nature. The switch structure uses four new keywords:

- The keyword **switch** starts the structure and is followed immediately by a test expression enclosed in parentheses.
- The keyword **case** is followed by one of the possible values for the test expression and a colon. You are not required to list the case value in any order. It is more efficient to list the most common case first, instead of the case with the lowest value.
- The keyword **break** optionally terminates a switch structure at the end of each case. The break statement bypasses the rest of the switch structure, and execution continues with any statement after the closing curly brackets of the switch structure. You can leave out the break statement in a switch structure. However, when you omit the break, if the program finds a match for the test variable, then all the statements within the switch statement from that point forward will execute.
- The keyword **default** optionally is used prior to any action that should occur if the test variable does not match any case.

This program uses a series of **if...else if** statements to perform decision making.

```
if (age == 13)
    System.out.println("Teenager!");
else if (age == 16)
    System.out.println("Keys, please?");
else if (age == 18)
    System.out.println("Rise to vote, sir!");
else if (age == 21)
    System.out.println("Full adult privileges");
else
    System.out.println("Too young to do anything.");
```

Here is the same program using a switch STATEMENT

```
switch (age)
{
    case 13: System.out.println("Teenager!");
             break;

    case 16: System.out.println("Keys, please?");
             break;

    case 18: System.out.println("Rise to vote, sir!");
             break;

    case 21: System.out.println("Full adult privileges");
             break;

    default: System.out.println("Too young to do anything.");
}
```

Lab 11 - ASSIGNMENT



Lab 11A - 70 points

OBJECTIVE

WAP that asks for a person's age from 10 to 25. Use the switch **statement shown above** and EXPAND IT to be able to handle all of these ages. You may add different output statements for other ages not indicated above, but every age must have an output.

The ages from 21 through 25 should all have the same output statement, **"Full adult privileges."**, but there should be only one **System.out.print** statement that covers every case.

FIELD SUMMARY

- **int age** – a person's age.

METHOD SUMMARY

- **main** – instantiate an instance of this class. Make method calls to `input` and `output`.
- **input** – declare a scanner object and read a person's age using an appropriate prompt.
- **output** – use a switch statement to display the appropriate results on the console screen.

SAMPLE KEYBOARD INPUT

Enter a person's age (Between 10 and 25): **18**

SAMPLE OUTPUT

Age 18: "Rise to vote sir!"



Lab 11B - 80 points

OBJECTIVE

WAP to calculate the cost of a telephone call when a code number and the length of the call is input. Output the 2 input values in a sentence along with the cost of the call. Test your program using a variety of code numbers and phone call lengths.

FIELD SUMMARY

- **int code** – the code number for the phone call.
- **int length** – the length of the phone call in minutes.
- **double cost** – the cost of the phone call.

METHOD SUMMARY

- **main** – instantiate an instance of this class. Make method calls to `input` and `output`.
- **input** – declare a `Scanner` object and read a code number and the length of a phone call using the prompt as shown below in the sample input.
- **process** - Use a switch statement to calculate the cost of the phone call.
- **output** – Display the appropriate results on the console screen.

SAMPLE KEYBOARD INPUT

<i>Code Table:</i>	<i>code</i>	<i>rate per minute</i>
	1	0.22
	2	0.14
	3	0.73
	4	1.12
	5	2.38

Enter a code number: 2
Enter the length of the call: 35

SAMPLE OUTPUT

Your code 2 call for 35 minutes costs \$4.90.



Lab 11C - 90 points

OBJECTIVE

Mr. Arthur Einstein, your high school physics teacher, wants a program for English to metric conversions. The input values will be a letter (**P** for pounds, **F** for feet, or **M** for miles) and a real number. The program is to make the appropriate conversions to Newtons, meters, and kilometers.

Your program will include only two instance fields (*char code* and *double units*) and must include a main method, a constructor, and input and output methods. Test your program using a variety of codes and values.

Code Table

Code	Rate Conversions
P	<i>Pounds to Newtons</i> : 1 Pound = 4.9 Newtons
F	<i>Feet to Meters</i> : 1 Foot = 0.30488 Meters
M	<i>Miles to Kilometers</i> : 1 Mile = 1.61 Kilometers

FIELD SUMMARY

- **char code** – the code to determine the conversion to be made.
- **double units** – a unit of measurement in English.

METHOD SUMMARY

- **main** – instantiate an instance of this class. Make method calls to `input` and `output`.
- **input** – declare a `Scanner` object and read a code (in the form of a `char`) and a number representing the value to be converted.
- **output** – use a switch statement to display the appropriate results on the console screen. Perform the calculations in the appropriate case statements.

SAMPLE KEYBOARD INPUT

<P – Pounds>
<F – Feet>
<M – Miles>
Make a selection: **M**
Enter a real number: **3.0**

SAMPLE OUTPUT

3.0 miles = 4.83 kilometers



Lab 11D - 100 points

OBJECTIVE

WAP that reads a series of months and days from a data file (“**lab11d.dat**”). For each month and day display the appropriate output based on the Zodiac table shown below. **Hint:** Switch on the months. Inside each `case` of the switch statement you will need an `if...else` statement to handle the range of days.

Sign	Symbol	Dates	Trait
<i>Aries</i>	The Ram	March 21 – April 20	Adventurous and energetic
<i>Taurus</i>	The Bull	April 21 – May 21	Patient and reliable
<i>Gemini</i>	The Twins	May 22 – June 21	Adaptable and Versatile
<i>Cancer</i>	The Crab	June 22 – July 22	Emotional and loving
<i>Leo</i>	The Lion	July 23 – August 21	Generous and warmhearted
<i>Virgo</i>	The Virgin	August 22 – September 23	Modest and shy
<i>Libra</i>	The Scales	September 24 – October 23	Diplomatic and urbane
<i>Scorpio</i>	The Scorpion	October 24 – November 22	Determined and forceful
<i>Sagittarius</i>	The Archer	November 23 – December 22	Optimistic and freedom-loving
<i>Capricorn</i>	The Goat	December 23 – January 20	Practical and prudent
<i>Aquarius</i>	The Water Carrier	January 21 – February 19	Friendly and humanitarian
<i>Pisces</i>	The Fishes	February 20 – March 20	Imaginative and sensitive

SAMPLE DATA FILE INPUT

6/13
12/25
5/11
1/28
4/10

SAMPLE OUTPUT

Genimi - The Twins: May 22 - June 21. Adaptable and Versatile.
Capricorn - The Goat: December 23 - January 20. Practical and prudent
Taurus - The Bull: April 21 - May 21. Patient and reliable.
Aquarius - The Water Carrier: January 21 - February 19. Friendly and humanitarian
Aries - The Ram: March 21 - April 20. Adventurous and energetic

