Switch Statement

Hui Chen

Department of Computer & Information Science

Brooklyn College

Objectives

- To implement selection control using switch statements (§3.13)

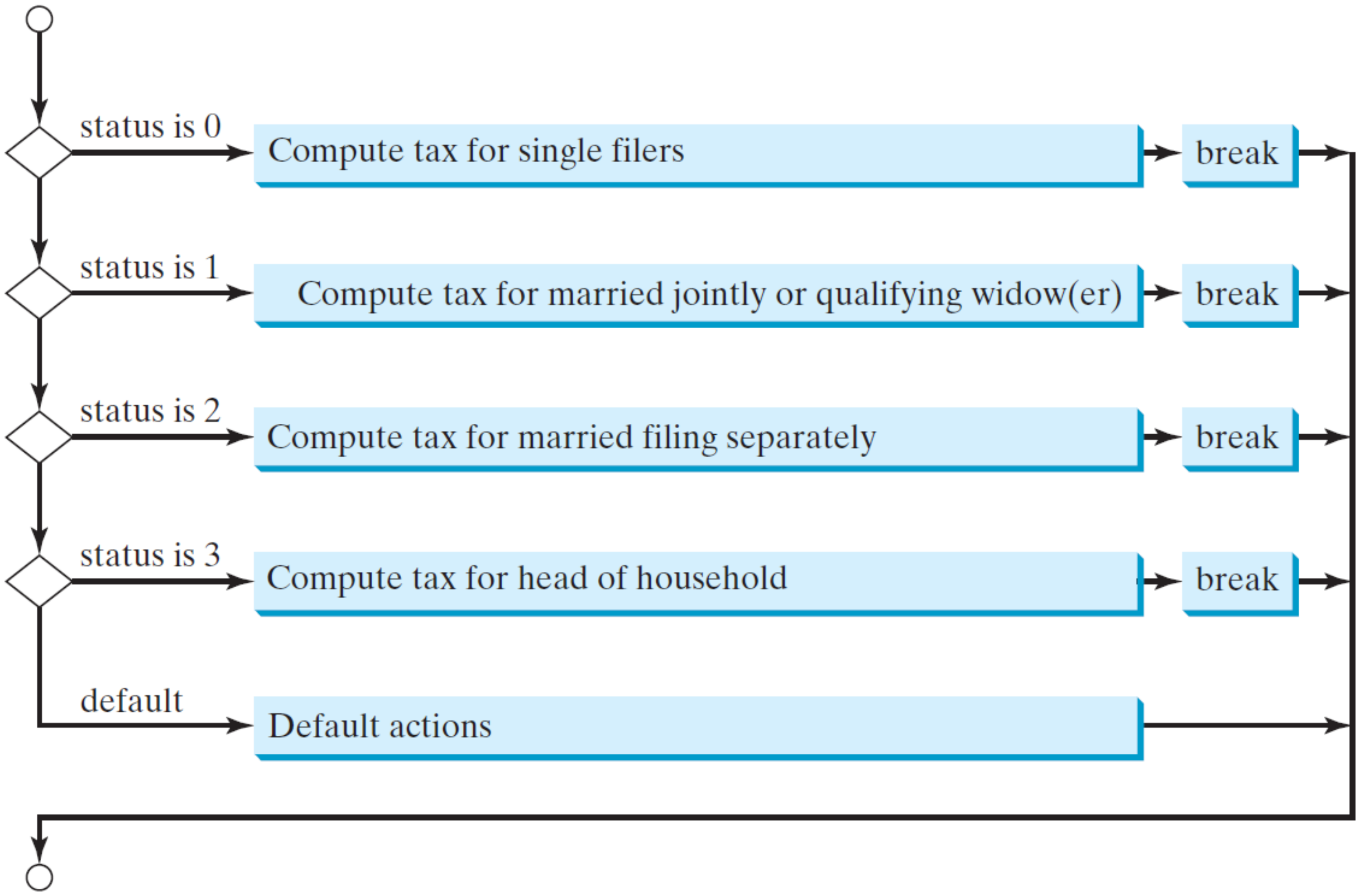
Outline

- Discussed
  - Boolean data type and Boolean expressions
  - If-statements (one-way, two-way, multi-way, and nested if-statements) and their flow charts
  - Common errors and pitfalls
  - 3 (“big”) programming problems (subtraction quiz, compute BMI, compute taxes)
  - Logical operators and two more (“big”) programming problems (LeapYear, Lottery)
- Switch statement

Switch Statement

```
switch (status) {
  case 0:  compute taxes for single filers;
           break;
  case 1:  compute taxes for married file jointly;
           break;
  case 2:  compute taxes for married file separately;
           break;
  case 3:  compute taxes for head of household;
           break;
  default: System.out.println("Errors: invalid status");
           System.exit(1);
}
```

Flow Chart of Switch Statement



status is 0
Compute tax for single filers
break
status is 1
Compute tax for married jointly or qualifying widow(er)
break
status is 2
Compute tax for married filing separately
break
status is 3
Compute tax for head of household
break
default
Default actions

Rules

The switch-expression must yield a value of char, byte, short, or int type and must always be enclosed in parentheses.

The value1, ..., and valueN must have the same data type as the value of the switch-expression. The resulting statements in the case statement are executed when the value in the case statement matches the value of the switch-expression. Note that value1, ..., and valueN are constant expressions, meaning that they cannot contain variables in the expression, such as 1 + x.

```
switch (switch-expression) {
  case value1:  statement(s)1;
         break;
  case value2: statement(s)2;
         break;
  …
  case valueN: statement(s)N;
         break;
  default: statement(s)-for-default;
}
```

The break Statement

The keyword break is optional, but it should be used at the end of each case in order to terminate the remainder of the switch statement. If the break statement is not present, the next case statement will be executed.

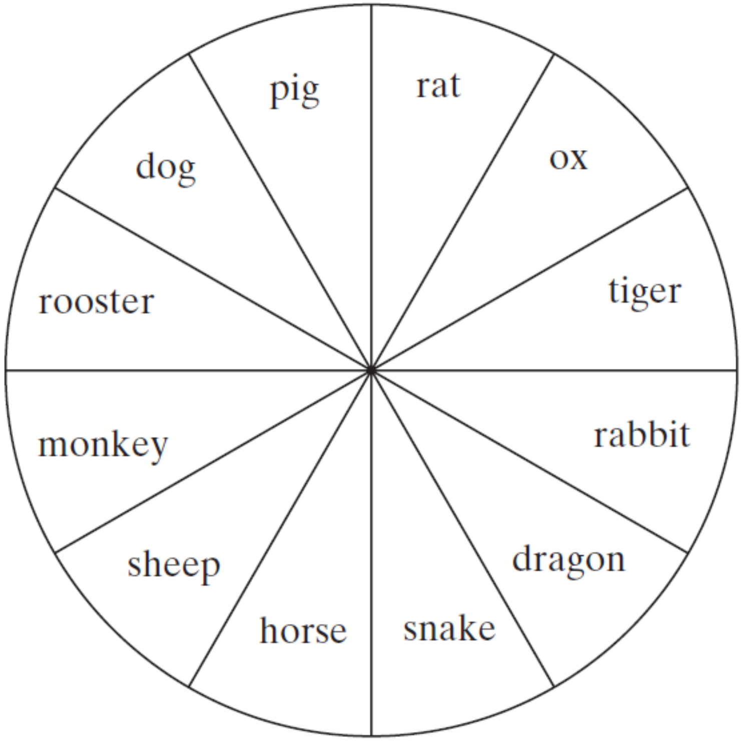
```
switch (switch-expression) {
  case value1:  statement(s)1;
           break;
  case value2: statement(s)2;
           break;
  …
  case valueN: statement(s)N;
           break;
  default: statement(s)-for-default;
}
```

The default case, which is optional, can be used to perform actions when none of the specified cases matches the switch-expression.

When the value in a **case** statement matches the value of the **switch-expression**, the statements *starting from this case* are executed until either a **break** statement or the end of the **switch** statement is reached.

Programming Problem. Display Chinese Zodiac

- Write a program that prompts the user to enter a year and displays the animal for the year.



`year % 12` =

0: monkey
1: rooster
2: dog
3: pig
4: rat
5: ox
6: tiger
7: rabbit
8: dragon
9: snake
10: horse
11: sheep

Questions?