PKG 05: ELEMENTARY PROGRAMMING 4, TYPE CONVERSIONS, and SELECTIONS

ELEMENTARY PROGRAMMING 4, Daniel Liang

Data Types and Type Conversions

- *float* and *double* for floating-point numbers
    - *double* = twice as big as *float*
    - *double,* double precision   more accurate   15-17 significant digits
    - *float,* single precision   less accurate   7-8 significant digits
    - float f = 5.9f; // double d
    - float f = 5.9F; // double D

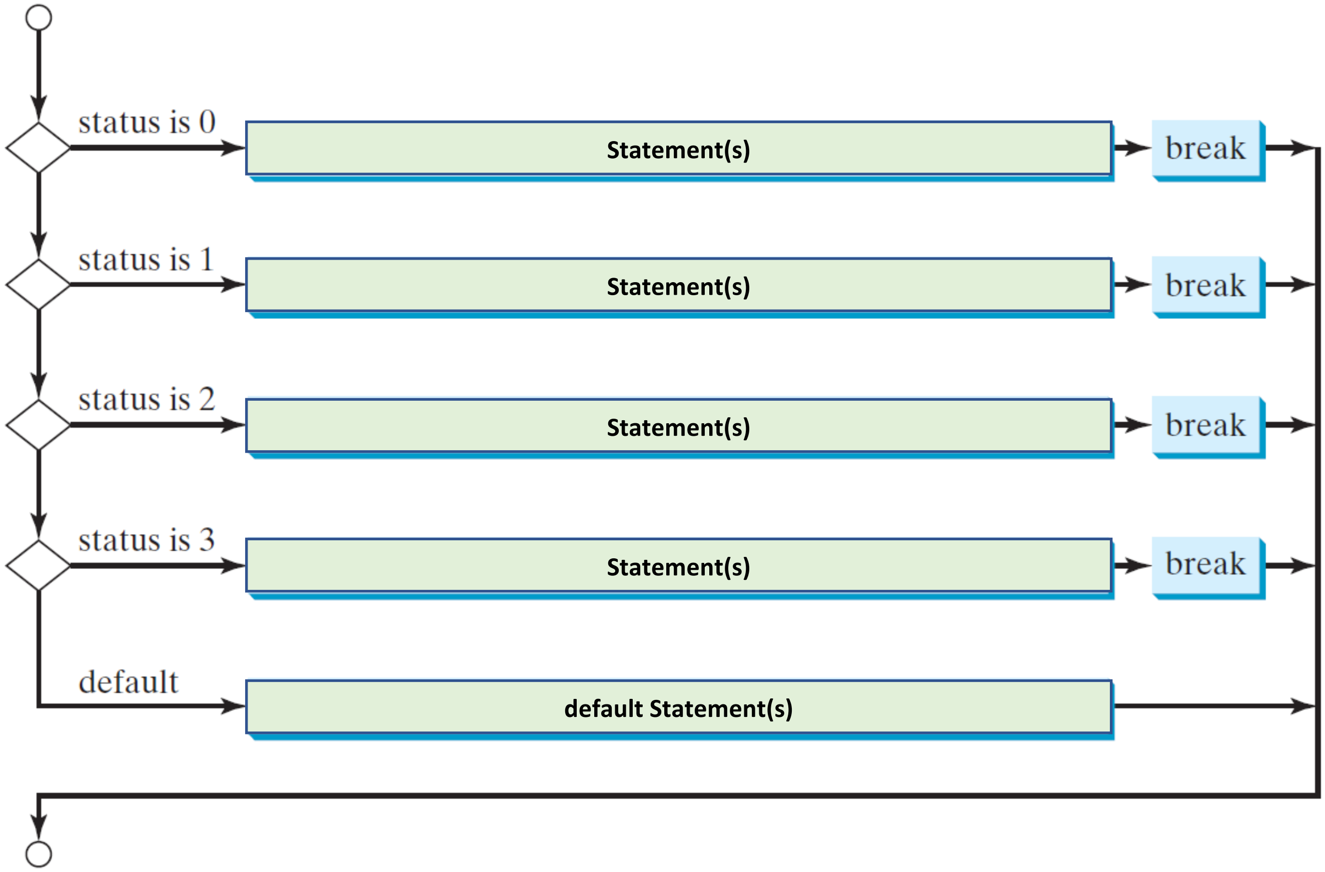
- *Casting* is an operation that converts a value of one data type into value of another data type: *widening* (automatically) and *narrowing* (explicitly)

```
public class Conversions {

    public static void main(String[] args) {

        // Integer to floating point automatically
        System.out.println(7 * 11.0);          // 77.0
        System.out.println(7.0 * 11.0);        // 77.0

        // Casting in parentheses, double into int
        System.out.println((int) 101.3);       // 101

        // int, floor
        System.out.println(7 / 11);            // 0
        System.out.println(11 / 7);            // 1

        // casted to float
        System.out.println((float) 7 / 11);    // 0.6363636
        System.out.println((float) 11 / 7);    // 1.5714285

        // casted to double
        System.out.println((double) 7 / 11);   // 0.6363636363636364
        System.out.println((double) 11 / 7);   // 1.5714285714285714
        System.out.format("Format %8f%n", (double) 11 / 7);    // Format 1.571429

        // casted to double
        System.out.println((double) (7 / 11)); // 0.0
        System.out.println((double) (11 / 7)); // 1.0
        System.out.format("Format %8f%n", (double) (11 / 7));  // Format 1.000000
    }
}
```

switch Statement

- The *switch-expression* must yield a value of *char*, *byte*, *short*, or *int* types and must always be enclosed in parentheses. (*more data types later*)
- Case data type must have the same data type as the value of the *switch-expression*.
- Case value cannot contain variables.
- 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 no break appears, the flow of control will fall through to subsequent cases until a break is reached.
- The *default* case, which is optional, can be used to perform actions when none of the specified cases matches the *switch-expression*. *default* does not need break.
- 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.

ELEMENTARY PROGRAMMING 4, TYPE CONVERSIONS, and SELECTIONS



```
public class Switch {

    public static void main(String[] args) {
        char grade = 'C';
        switch (grade) {
            case 'A':
                System.out.println("Excellent!");
                break;
            case 'B':
            case 'C':
                System.out.println("Well done");
                break;
            case 'D':
                System.out.println("You passed");
            case 'F':
                System.out.println("Better try again");
                break;
            default:
                System.out.println("Invalid grade");
        }
        System.out.println("Your grade is " + grade);
    }
}
```

```
Well done
Your grade is C
```

CHALLENGE:

Rewrite the Letter Grade program using switch statements
Make the program:

- Print: “Good work! Keep it up” if A, B, or C.
- Print: “See your professor. Work harder.” if D or F.
- Only use switch statements