CT60A2411
Conditional selection statements: Week 3

Ashok Kumar Veerasamy, PhD

Learning objectives: This week

- Conditional / Selection statements
- Loops
- Single dimensional arrays

At the conclusion of this lecutre, students will be able to know using if-else and looping statements for execution of decision-based coding In addition, know how to use data structure by using Java's arrays for coding.

Selection statements

Java has two selection statements namely;

- simple if , if-else and nested if statements
- switch statement

To execute decision-based code (including loops) needs comparison and logical operators with aforenoted decision and looping statements.

Operator	*Name*
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
==	equal to
!=	not equal to

Comparison operators

Operator	*Name*
!	not
&&	and
||	or
^	exclusive or

Logical operators

- **If statement**

```
if (booleanExpression) {
  statement(s);
}
```

```
if (radius >= 0) {
  area = radius * radius * 3.14159;
  System.out.println("The area of the circle is:"+ area);
}
```

```
if (booleanExpression) {
  statement(s);
}
else {
  statement(s);
}
```

```
if (radius >= 0) {
  area = radius * radius * 3.14159;

  System.out.println("The area of the circle is:"+ area);}
else {
  System.out.println("Negative input");
}
```

- **Nested if statement**

```
if (booleanExpression) {
 statement(s);
}
else if{
 statement(s);
}

else if{
 statement(s);
}
--
else {
 statement(s);
}
```

Why no braces→{ } here?

```
if (score >= 90.0)
  grade = 'A';
else
  if (score >= 80.0)
    grade = 'B';
  else
    if (score >= 70.0)
      grade = 'C';
    else
      if (score >= 60.0)
        grade = 'D';
      else
        grade = 'F';
```

Equivalent

```
if (score >= 90.0)
  grade = 'A';
else if (score >= 80.0)
  grade = 'B';
else if (score >= 70.0)
  grade = 'C';
else if (score >= 60.0)
  grade = 'D';
else
  grade = 'F';
```

Here **even** is a boolean datatype variable

```
if (even == true)
  System.out.println(
    "It is even.");
```

(a)

Equivalent

```
if (even)
  System.out.println(
    "It is even.");
```

(b)

```
if (number % 2 == 0)
  even = true;
else
  even = false;
```

(a)

Equivalent

```
boolean even
  = number % 2 == 0;
```

(b)

- **What will be the output of the program below?**

```
public class Compare
{
   public static void main (String[] args)
   {
      int num1 = 10, num2 = 10;
      Integer rnum1 = new Integer(20);// int rnum1 = 20;
      Integer rnum2 = new Integer(20);// int rnum2 = 20;
      if( num1 == num2)
         System.out.println ("num1 is equal to num2");
      if( rnum1 == rnum2)
         System.out.println ("rnum1 is equal to rnum2");
      else
         System.out.println("rnum1 not equal to rnum2");

      if( rnum1.intValue() == rnum2.intValue())
         System.out.println("objects have same value");
      System.out.println();
   }
}
```

- **The code given below did not print correct statements when the input for age is 17 or 21 for example. Rewrite it by using logical operators.**

```
import java.util.Scanner;

public class exampleIf {
        public static void main(String[] args) {
         Scanner input = new Scanner(System.in);
         System.out.print("Enter your age: ");
         int age = input.nextInt();
         if (age<=0){
                 System.out.print("Input is wrong");
         }
         else
                if (age<=10)
                if (age<19) {
                        System.out.print("Not eligible
should be at least 20 years old");

         }
                else {
                        System.out.print("Eligible");
                }
}
}
```

- **Switch statement**

```
System.out.print("Enter your option");
int option = input.nextInt();

if (option==1) {
        //print vehicle details
        // your code here
}
else if (option==2){
        //Add vehicle details
        // your code here
}

else if (option==3) {
//Delete vehicle details
// your code here
}

else {
// the option ends Bye code here
}
```

```
switch (option) {
        case 1:
        //print vehicle details
        // your code here;
        break;
        case 2:
        //Add vehicle details
        // your code here
        break;
        case 3:
        //Delete vehicle details
        // your code here
        break;
        default:
        // the option ends Bye code here
}
```

So, what is the difference and is break statement at the end of each case is compulsory?

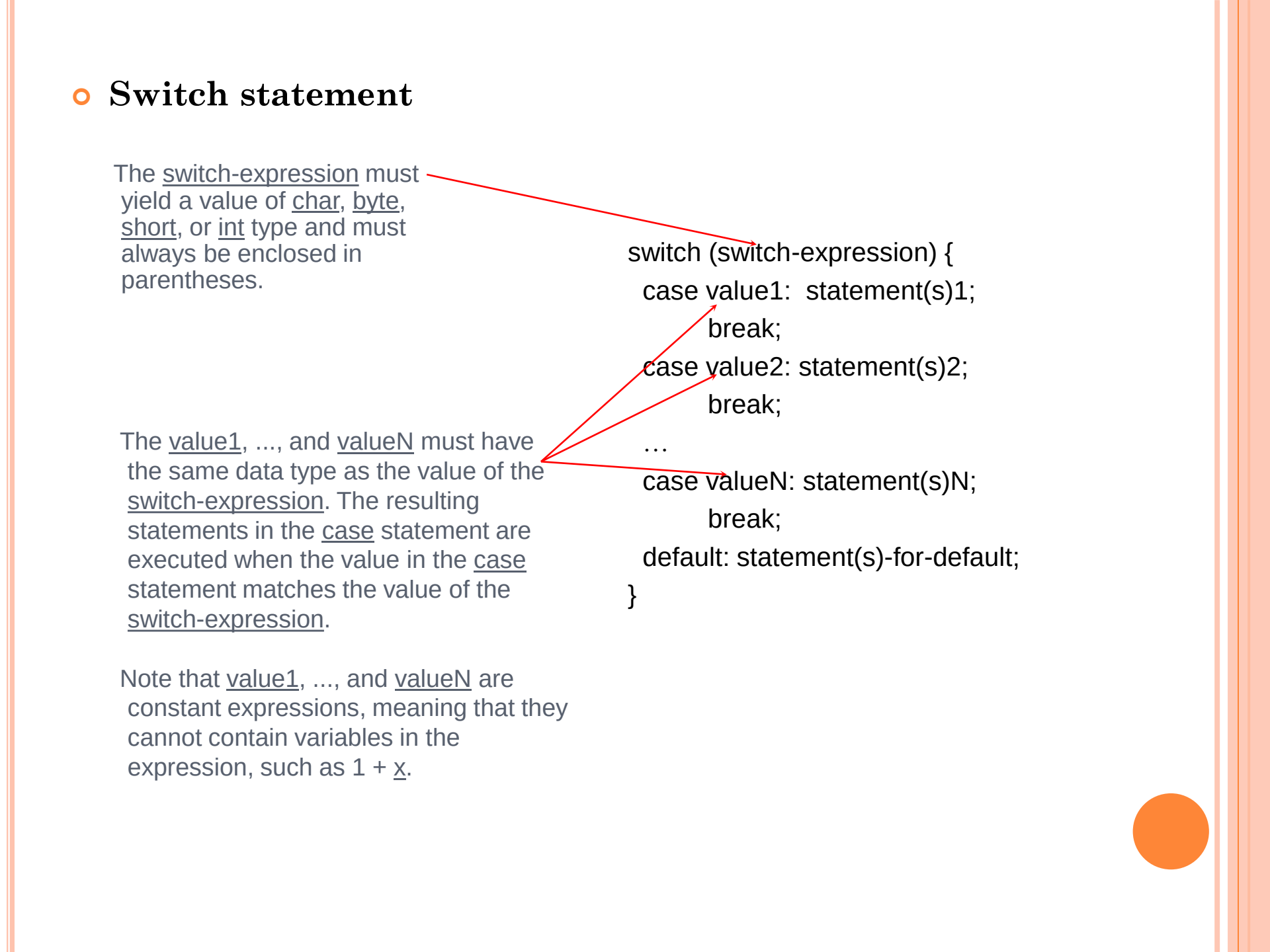
Switch statement

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;
}
```



Switch 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.

The case statements are executed in sequential order, but the order of the cases (including the default case) does not matter. However, it is good programming style to follow the logical sequence of the cases and place the default case at the end.

- **Formatted output**

```
Scanner sc = new Scanner(System.in);
System.out.println("PI value is :" +Math.PI);
System.out.printf("PI value is :%.3f" ,Math.PI); // math library
System.out.printf("\n"); \\ line space
int x = 14500;
System.out.printf(" x value is :%d\n" ,x); // integer %d

System.out.println("Enter kilograms:");
double kg = sc.nextDouble();
double p = kg *2.20462;

System.out.printf("\nKilograms:%.2f and equvalent pounds is:%.1f
",kg,p);

}
}
```

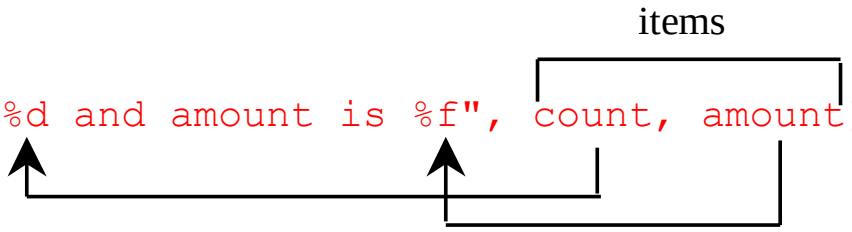
```
PI value is :3.141592653589793
PI value is :3,142
x value is :14500
Enter kilograms:
3

Kilograms:3,00 and equvalent pounds is:6,6
```

Formatted output

Specifier	Output	Example
%b	a boolean value	true or false
%c	a character	'a'
%d	a decimal integer	200
%f	a floating-point number	45.460000
%e	a number in standard scientific notation	4.556000e+01
%s	a string	"Java is cool"

```
int count = 5;
double amount = 45.56;
System.out.printf("count is %d and amount is %f", count, amount);
```




```
display            count is 5 and amount is 45.560000
```