

CSE3040 Java Language

Lecture #05

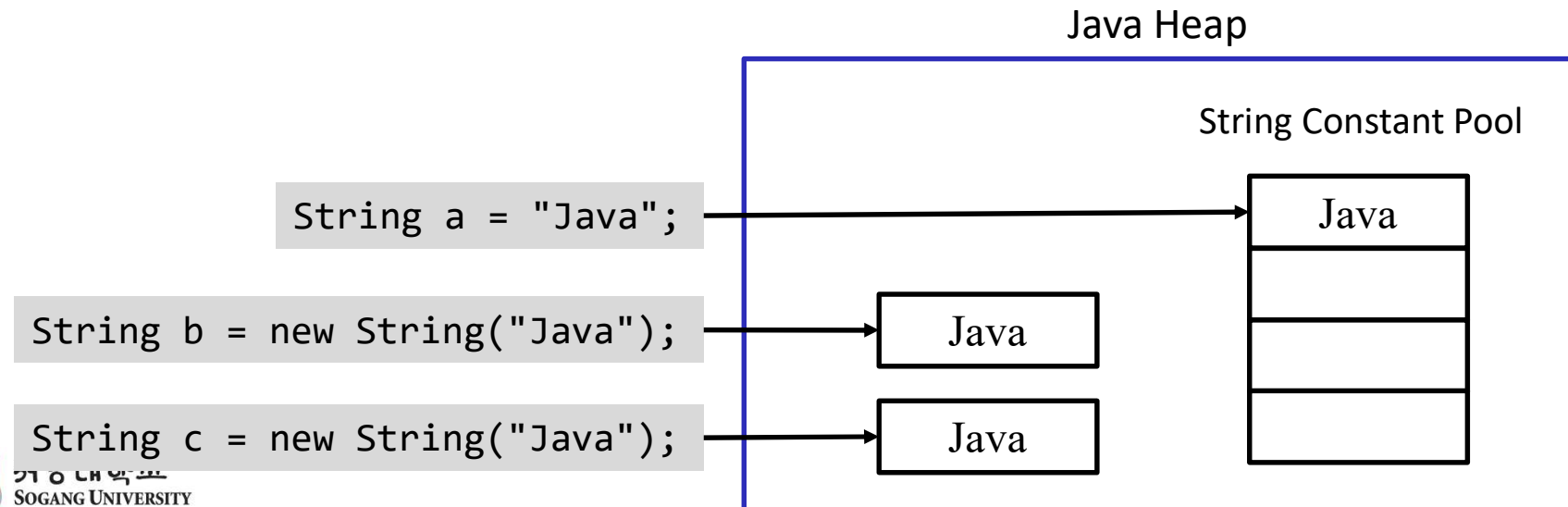
Dept. of Computer Engineering,
Sogang University

This material is based on lecture notes by Prof. Juho Kim. Do not post it on the Internet.

1.5. Strings

- Assigning a string literal vs. Creating a new String object
 - There are two ways to assign a string to a String type variable.

```
String a = "Java";  
String b = new String("Java");
```
 - A string literal is directly assigned to variable a.
 - A new String object whose value is "Java" is created and assigned to b.
- When a string literal is used, it is stored in a memory region called "String Constant Pool".
- When a String object is created, it is allocated a separate memory space.



1.5. Strings

```
class dtStrBiGyo {
    public static void main(String[] args)
    {
        String s1 = "Java";
        String s2 = "JAVA";

        if (s1 == "Java")          System.out.println("same");
        else                        System.out.println("different");

        //compare "JAVA" with "JAVA". Result : different
        if (s1.toUpperCase() == s2) System.out.println("same");
        else                        System.out.println("different");

        String su = s1.toUpperCase(); // su == "JAVA"

        // compare s2 with su. Result : same
        if (s2.equals(su))          System.out.println("same");
        else                        System.out.println("different");
    }
}
```

1.5. Strings

- More String methods
 - boolean startsWith(String str)
 - boolean endsWith(String str)
 - boolean contains(CharSequence str)
 - int indexOf(String str)
 - int lastIndexOf(String str)
 - int indexOf(String str, int fromIndex)
 - int lastIndexOf(String str, int fromIndex)
 - String replace(CharSequence oldString, CharSequence newString)
 - String toUpperCase()
 - String toLowerCase()
 - String trim()

1.6. Inputs and Outputs

- Reading user input
 - The `nextLine` method of class `Scanner` reads a line from standard input.

```
Scanner in = new Scanner(System.in);  
System.out.println("What is your name?");  
String name = in.nextLine();
```

- Other methods
 - `next`: read one word separated by a space.
 - `nextInt`: read an integer
 - `nextDouble`: read a double-type number.

1.6. Inputs and Outputs

- Formatted output
 - System.out.print: does not move to the next line after printing the string.
 - System.out.println: moves to the next line after printing the string.
 - System.out.printf: formatted output (similar to printf in C).

```
System.out.printf("%8.2f", 1000.0/3.0);
```

```
System.out.printf("Hello, %s. Next year, you will be %d.\n", name, age);
```

- Conversion character
 - d: integer
 - x: hexadecimal number
 - f: floating point
 - c: character
 - s: string
 - b: boolean

1.6. Inputs and Outputs

- Flags
 - +: shows the sign
 - -: left align the number
 - 0: pad zeros in front of the number
 - (: use parenthesis for negative numbers instead of negative sign.
 - ,: use commas for number groups

```
System.out.printf("%,+2f", 100000.0/3.0);  
// the results is +33,333.33
```

- String.format: instead of printing to the standard output, return a String.

```
String message = String.format("Hello, %s. Next year, you'll be %d\n", name, age);
```

1.7. Control Flow

- Control flow statements
 - if statement
 - for loop
 - while loop
 - do-while loop
 - switch block
 - break
 - continue
- Similar to C/C++

1.7. Control Flow

- If block

```
if (condition) {  
    (code block 1)  
}  
else {  
    (code block 2)  
}
```

- If condition is true, (code block 1) will be executed.
- If condition is false, (code block 2) will be executed.
- else (code block 2) is omissible.
- If a code block is a single statement, { } can be omitted.

1.7. Control Flow

- Example: Generate a random number n , and check whether n is positive or negative.

```
class If1 {  
    public static void main(String[] args)  
    {  
        // n : -5 ~ 5 random number(integer)  
        int n = (int)( 11*Math.random() ) - 5;  
        System.out.println("Random number : " + n);  
  
        if (n > 0) System.out.println("positive");  
        else System.out.println("0 or negative");  
    }  
}
```

1.7. Control Flow

- Example: If the random number is less than 5, print n and n^2.

```
class If2 {  
    public static void main(String[] args)  
    {  
        // n : 0 ~ 10 random number(integer)  
        int n = (int)( 11*Math.random() );  
        System.out.println("random number : " + n);  
  
        if (n<=5)  
        {  
            System.out.println("n    : " + n);  
            System.out.println("n*n : " + n*n);  
        }  
        System.out.println();  
        System.out.println("end");  
    }  
}
```

1.7. Control Flow

- if-else if-else

```
if (condition 1) {  
    (code block 1)  
}  
else if (condition 2) {  
    (code block 2)  
}  
else {  
    (code block 3)  
}
```

- If condition 1 is true, (code block 1) will be executed.
- If condition 1 is false and condition 2 is true, (code block 2) will be executed.
- If both condition 1 and condition 2 are false, (code block 3) will be executed.

1.7. Control Flow

- Example: Generate a random number and print where the number belongs.

```
class If3 {  
    public static void main(String[] args)  
    {  
        // n : -100 ~ 100 random number(integer)  
        int n = (int)( 200*Math.random() ) - 100;  
        System.out.println("Random number : " + n);  
  
        if (n>=100)  
            System.out.println("more than 100");  
        else if (n>0)  
            System.out.println("0 ~ 100");  
        else if (n>-100)  
            System.out.println("-100 ~ 0 ");  
        else  
            System.out.println("less than -100");  
    }  
}
```

1.7. Control Flow

- for loop

```
for (counter_initialization; condition; update_counter)  
    statement
```

- The first slot of the for statement usually holds the counter initialization.
- The second slot gives the condition which will be tested before each new pass through the loop.
- The third slot explains how to update the counter

- `for(i = 1 ; i < 10 ; i++) <statement>`

- For $i = 1, 2, 3, \dots, 9$, `<statement>` is repeatedly executed 9 times.
If $i == 10$, for loop is terminated.

- `for(i = 2 ; i < 10 ; i+=3) <statement>`

- For $i = 2, 5, 8$, `<statement>` is repeatedly executed 3 times.
If $i == 11$, for loop is terminated.

1.7. Control Flow

- Example: sum of numbers

```
class For {  
    public static void main(String[] args)  
    {  
        // i, sum (integer type variable)  
        int i, sum=0;  
  
        // for loop of sum 1 ~ 10  
        for (i=1 ; i<=10 ; i++)  
        {    System.out.print(i + "  ");  
            sum += i;  
        }  
        System.out.println();           // new line  
        System.out.println("1 ~ 10 sum : " + sum);  
    }  
}
```

1.7. Control Flow

- Endless for loop
 - If condition is always true or is omitted, for loop is endlessly repeated.
 - Example
 - `for(i = 1 ; true ; i++) (statement)`
 - `for(i = 10 ; ; i--) (statement)`
 - `for(; ;) (statement)`
 - In order to terminate a loop, we can use `break`.

1.7. Control Flow

- While loop

- As long as condition is true, <statement> is repeatedly executed.

```
while (condition)
    statement
```

- Example: sum of integers from 1 to 100.

```
class While {
    public static void main(String[] args)
    {
        int    i=0;
        long sum=0;           // sum init.

        while ( i<=100 )      // i<=100
        {
            sum += i;         // sum=sum+i
            i++;              // i=i+1
        }
        System.out.println("sum(1 ~ 100) : " + sum);
    }
}
```

1.7. Control Flow

- Do-while loop

- A while loop tests at the top. Therefore, the code in the block may never be executed.
- do ~ while loop executes the block and only then tests the condition.
- It then repeats the block and retests the condition, and so on.

```
do  
    <statement>  
while  
    (condition);
```

1.7. Control Flow

- Example: Generate random number n, and find sum of 1 to n.

```
class DoWhile {
    public static void main(String[] args)
    {
        int    i=0, n;
        long   sum=0;
        // n : 10 ~ 15 random number(integer)
        n = (int)( 6*Math.random() ) + 10;
        System.out.println("Random number : " + n);
        do {
            sum += i;
            i++;
        }
        while (i<=n);
        System.out.println("\n i          : " + i);
        System.out.println(" 1 ~ n sum : " + sum);
    }
}
```

1.7. Control Flow

- Switch block
 - Execution starts at the case label that matches the value on which the selection is performed and continues until the next break or the end of the switch.
 - If none of the case labels matches, then the default clause is executed, if it is present

```
switch (choice)
{
    case 1 : <statement 1> break;
    case 2 : <statement 2> break;
    case 3 : <statement 3> break;
    .
    .
    .
    default : <statement N> break;
}
```

1.7. Control Flow

- Switch block example

```
class Switch1 {  
    public static void main(String[] args)  
    {  
        int n;  
        // n : 10 ~ 20 random number(integer)  
        n = (int)( 11*Math.random() ) + 10;  
        System.out.println("Random number : " + n);  
        System.out.println("\nn % 5 : " + (n % 5) + "\n");  
        switch (n % 5)  
        {  
            case 0: System.out.println("remainder 0");  
            case 1: System.out.println("remainder 1");  
            case 2: System.out.println("remainder 2");  
            default : System.out.println("remainder 3 or 4");  
        }  
    }  
}
```

1.7. Control Flow

- Use of **break** statements in a switch block

```
class Switch2 {  
    public static void main(String[] args)  
    {  
        int    n;  
        // n : 10 ~ 20 random number(integer)  
        n = (int)( 11*Math.random() ) + 10;  
        System.out.println("Random number : " + n);  
        System.out.println("n % 5 : " + (n % 5) + "\n");  
        switch (n % 5)  
        {  
            case 0: System.out.println("remainder 0"); break;  
            case 1: System.out.println("remainder 1"); break;  
            case 2: System.out.println("remainder 2"); break;  
            default : System.out.println("remainder 3 or 4"); break;  
        }  
    }  
}
```

1.7. Control Flow

- break: move out of the loop

```
class Break1 {
    public static void main(String[] args)
    {
        int    n, sum=0;

        while ( true )    // endless while loop
        {
            n = (int)( 11*Math.random() ) + 10;
            System.out.print(n + " ");
            if (n == 15)    // if n is 15, break out of a loop
                break;
            sum += n;    // sum = sum + n
        }
        System.out.println("\nsum : " + sum);
    }
}
```

1.7. Control Flow

- **continue**
 - transfers control to the header of the innermost enclosing loop

```
while(sum < goal)
{
    ...
    if (n < 0) continue;
    ...
}
```

```
for(count=0; count<100; count++)
{ ...
    if (n<0) continue;
    ...
}
```

- If $n < 0$, the `continue` statement jumps immediately to the loop header, skipping the remainder of the current iteration.

1.7. Control Flow

- continue: Example

```
class Continue {
    public static void main(String[] args)
    {
        int n=1;

        while ( n != 0 )
        {
            n = (int)( 5*Math.random() ) - 2;
            System.out.print(n + " ");
            if (n < 0)           // if n is negative
            {
                System.out.println("n is negative number!");
                continue;
            }
            System.out.println("square root : " + Math.sqrt(n));
        }
    }
}
```

Programming Lab #05

05-01. String Comparison

- What will be printed on the display when you execute this program?
- Guess first, and then run this program and see the result for yourself.

```
public class Ex05_01 {  
    public static void main(String[] args) {  
        String s1 = "Java";  
        String s2 = "JAVA";  
  
        if(s1 == "Java") System.out.println("same");  
        else System.out.println("different");  
  
        if(s1.toUpperCase() == s2) System.out.println("same");  
        else System.out.println("different");  
  
        String su = s1.toUpperCase();  
        if(s2.equals(su)) System.out.println("same");  
        else System.out.println("different");  
    }  
}
```

05-02. Input and Output

- Write a Java program that asks for name of the user and prints a string.
 - First, the program should print "What is your name?".
 - Then, the program should wait for user input and store it in a variable *name*.
 - Then, the program should print "Hello! *name*".

05-02. Input and Output

- Write a Java program that asks for name of the user and prints a string.
 - First, the program should print "What is your name?".
 - Then, the program should wait for user input and store it in a variable *name*.
 - Then, the program should print "Hello! *name*".

```
import java.util.Scanner;
public class Ex05_02 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("What is your name?");
        String name = in.nextLine();
        String message = String.format("Hello %s!", name);
        System.out.println(message);
        in.close();
    }
}
```

05-03. if statement

- Write a Java program that does the following.
 - Draw a random integer n in the range $[-200, 200)$, and print "random number: n ".
 - On the second line,
 - If $n > 100$, print "more than 100".
 - If $0 \leq n \leq 100$, print "0 ~ 100".
 - If $-100 \leq n < 0$, print "-100 ~ -1".
 - If $n < -100$, print "less than -100".

05-03. if statement

- Write a Java program that does the following.
 - Draw a random integer n in the range $[-200, 200)$, and print "random number: n ".
 - On the second line,
 - If $n > 100$, print "more than 100".
 - If $0 \leq n \leq 100$, print "0 ~ 100".
 - If $-100 \leq n < 0$, print "-100 ~ -1".
 - If $n < -100$, print "less than -100".

```
public class Ex05_03 {  
    public static void main(String[] args) {  
        int n = (int)(400 * Math.random()) - 200;  
        System.out.println("random number: " + n);  
        if(n > 100) System.out.println("more than 100");  
        else if(n >= 0) System.out.println("0 ~ 100");  
        else if(n > -100) System.out.println("-100 ~ -1");  
        else System.out.println("less than -100");  
    }  
}
```

05-04. for loop

- Write a Java program that does the following.
 - On the first line, print numbers from 1 to 10.
 - On the second line, print the sum of numbers from 1 to 10.
 - Use **for loop** to achieve the goal.

05-04. for loop

- Write a Java program that does the following.
 - On the first line, print numbers from 1 to 10.
 - On the second line, print the sum of numbers from 1 to 10.
 - Use **for loop** to achieve the goal.

```
public class Ex05_04 {  
    public static void main(String[] args) {  
        int i, sum=0;  
        for(i=1; i<=10; i++) {  
            System.out.print(i + " ");  
            sum += i;  
        }  
        System.out.println();  
        System.out.println("sum: " + sum);  
    }  
}
```

05-05. while loop

- Write a Java program that does the following.
 - Print the sum of numbers from 1 to 100.
 - Use **while loop** to achieve the goal.

05-05. while loop

- Write a Java program that does the following.
 - Print the sum of numbers from 1 to 100.
 - Use **while loop** to achieve the goal.

```
public class Ex05_05 {  
    public static void main(String[] args) {  
        int i=0, sum=0;  
        while( i <= 100 ) {  
            sum += i;  
            i++;  
        }  
        System.out.println("sum(1-100): " + sum);  
    }  
}
```

05-06. do-while loop

- Write a Java program that does the following.
 - Print the sum of numbers from 1 to 100.
 - Use **do-while loop** to achieve the goal.

05-06. do-while loop

- Write a Java program that does the following.
 - Print the sum of numbers from 1 to 100.
 - Use **do-while loop** to achieve the goal.

```
public class Ex05_06 {  
    public static void main(String[] args) {  
        int i=0, sum=0;  
        do {  
            sum += i;  
            i++;  
        } while( i <= 100 );  
        System.out.println("sum(1-100): " + sum);  
    }  
}
```

05-07. switch statement

- Write a Java program that does the following.
 - Draw a random number from 1 to 100 and print the number.
 - Divide the number by 5 and get the remainder `r`. Print "n % 5: `r`"
 - If the remainder is 0, print "remainder is zero."
 - Else if the remainder is 1, print "remainder is one."
 - Else if the remainder is 2, print "remainder is two."
 - Otherwise, print "remainder is three or four."
- use **switch statement** to achieve the goal.

05-07. switch statement

```
public class Ex05_07 {  
    public static void main(String[] args) {  
        int n = (int)(Math.random() * 100) + 1;  
        System.out.println("random number: " + n);  
        System.out.println("n % 5: " + (n%5));  
        switch(n % 5) {  
            case 0:  
                System.out.println("remainder is zero."); break;  
            case 1:  
                System.out.println("remainder is one."); break;  
            case 2:  
                System.out.println("remainder is two."); break;  
            default:  
                System.out.println("remainder is three or four.");  
        }  
    }  
}
```

05-08. break & continue

- Write a Java program that does the following.
 - Draw a random number from 0 to 99 and print the number.
 - If 0 is drawn, finish the program.
 - If a number among 10, 20, 30, 40, 50, 60, 70, 80, 90 is drawn, don't print the number and repeat.
 - Otherwise, print the number and repeat.
 - Use the keyword **break** and **continue** to achieve the goal.

05-08. break & continue

- Write a Java program that does the following.
 - Draw a random number from 0 to 99 and print the number.
 - If 0 is drawn, finish the program.
 - If a number among 10, 20, 30, 40, 50, 60, 70, 80, 90 is drawn, don't print the number and repeat.
 - Otherwise, print the number and repeat.
 - Use the keyword **break** and **continue** to achieve the goal.

```
public class Ex05_08 {  
    public static void main(String[] args) {  
        int n;  
        while(true) {  
            n = (int)(Math.random() * 100);  
            if(n == 0) break;  
            if(n % 10 == 0) continue;  
            System.out.println(n);  
        }  
    }  
}
```

End of Class



Instructor office: AS818A

Email: jso1@sogang.ac.kr