CS1101 Programming and Problem Solving

Dr. Gina Bai

Spring 2023

Logistics

- **ZY-4B** on **zyBook > Assignments**
  - Due: **Wednesday, Feb 22**, at 11:59pm
- **PA05 – W, A, B** on **zyBook > Chap 11**
  - Due: **Thursday, Feb 23**, at 11:59pm

Recap – while Loops

Q: What is the output of the following code?

```
int b = 1;
while (b < 4){
    System.out.print(b + " ");
    b += 1;
}
```

A. 1

B. 1 2 3 ✓

C. 1 2 3 4

D. 1 1 1 1 1 1 1 1 1 1 …

```
public static void mystery(int x) {
    int y = 0;
    while (x % 2 == 0) {
        ++y;
        x = x / 2;
    }
    System.out.println(x + " " + y);
}
```

Q: What is the output of `mystery(42)`?

21 1

→ x = 42, y = 0

→ is x % 2 == 0? Yes

→ y = 1, x = 21

→ is x % 2 == 0? No

Q: What is the output of `mystery(40)`?

5 3

→ x = 40, y = 0

→ is x % 2 == 0? Yes, y = 1, x = 20

→ is x % 2 == 0? Yes, y = 2, x = 10

→ is x % 2 == 0? Yes, y = 3, x = 5

→ is x % 2 == 0? No

do-while Loop

zyBook Chap 5.8

do-while Loops

- **Do**/execute the <controlled stmt(s)> **once**, **then check** the <condition>, **while** the <condition> is **`true`**, execute the <controlled stmt(s)>

- Hence, the <controlled stmt(s)> in a do-while loop can be executed **AT LEAST ONCE**.

```
do {
     <controlled stmt(s)>;
} while (<condition>);

<statement>;
```

```java
public class MultiplesOfThree {
    public static void main (String[] args) {

        int val = 3;

        /*
         * 1) Execute the controlled stmts
         *    - print out the value
         *    - increment val by 3
         * 2) Check if the condition (val <= 20) is true
         *    If yes, execute the controlled stmts
         *    If no, skip the while loop
         */
        do {
            System.out.println(val);
            val += 3;
        } while (val <= 20);
        System.out.println("Done.");
    }
}
```

val = 3

print out 3
val += 3 // 6

val <= 20 ?
True
print out 6
val += 3 // 9

val <= 20 ?
True
print out 9
val += 3 // 12

val <= 20 ?
True
print out 12
val += 3 // 15

val <= 20 ?
True
print out 15
val += 3 // 18

val <= 20 ?
True
print out 18
val += 3 // 21

val <= 20 ?
False
print out Done.

Q: What's the exact output?

```
public class DoWhileExample {
    public static void main(String[] args) {

        int a = 57;
        do {
            System.out.print(a % 5);
            a = a / 5;
        } while(a > 0);
    }
}
```

Output:

212

a = 57

print out 2 // 57 % 5
a = 57 / 5   // 11

a > 0 ?
True
print out 1 // 11 % 5
a = 11 / 5   // 2

a > 0 ?
True
print out 2 // 2 % 5
a = 2 / 5   // 0

a > 0 ?
False

Q: What's the exact output?

```
public class DoWhileExample {
    public static void main(String[] args) {

        int a = 1;
        do {
            a++;
            System.out.print(a);
            ++a;
        } while (a <= 10);
    }
}
```

Output:
246810

a = 1

a += 1 // a = 2
print out 2
a += 1 // a = 3

a < = 10 ?
True
a += 1 // a = 4
print out 4
a += 1 // a = 5

a < = 10 ?
True
a += 1 // a = 6
print out 6
a += 1 // a = 7

a < = 10 ?
True
a += 1 // a = 8
print out 8
a += 1 // a = 9

a < = 10 ?
True
a += 1 // a = 10
print out 10
a += 1 // a = 11

a < = 10 ?
False

while Loops vs. do-while Loops

Q: Assuming `x` is a random integer from the user

- Rewrite the following while loop with a do-while loop

```
while (x <= 10) {
    System.out.print(x);
    ++x;
}
System.out.print("Done.");
```

```
do {
    if (x <= 10) {
        System.out.print(x);
        ++x;
    }
} while (x <= 10);
System.out.print("Done.");
```

- Rewrite the following do-while loop with a while loop

```
do {
    System.out.print(x);
    ++x;
} while (x <= 10);
System.out.print("Done.");
```

```
System.out.print(x);
++x;
while (x <= 10) {
    System.out.print(x);
    ++x;
}
System.out.print("Done.");
```

while Loops vs. do-while Loops

- In a **while** loop,
  - the condition is **tested** at the **beginning** of the loop
  - executes the controlled statements **zero or many** times

- In a **do-while** loop,
  - the condition is **tested** at the **end** of the loop
  - executes the controlled statements **at least once**

Coding Practice – Part 1

Write a program called **WhileLoops** that

- prompts the user for **two integers** (first value and second value), and
- uses a **while loop** to print out **all integers** between these two integers, one per line.

```
$ java WhileLoops
First value (int): 6
Second value (int): 10
6
7
8
9
10

$ java WhileLoops
First value (int): -3
Second value (int): -7
-3
-4
-5
-6
-7
```

Sample Solution

```
$ java WhileLoops
First value (int): 6
Second value (int): 10
6
7
8
9
10

$ java WhileLoops
First value (int): -3
Second value (int): -7
-3
-4
-5
-6
-7
```

```java
import java.util.Scanner;

public class WhileLoops {
    public static void main (String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("First value (int): ");
        int firstVal = input.nextInt();
        System.out.print("Second value (int): ");
        int secondVal = input.nextInt();

        int val = firstVal;

        if (firstVal < secondVal) {
            while(val <= secondVal) {
                System.out.println(val);
                ++val;
            }
        } else {
            while(val >= secondVal) {
                System.out.println(val);
                --val;
            }
        }
    }
}
```

Coding Practice – Part 2

Write a program called **DoWhileLoops** that

- prompts the user for **an integer**, and
- uses a **do-while loop** to print out all **even integers** between 0 and the given integer, one per line.

```
$ java DoWhileLoops
Enter an integer: 12
0
2
4
6
8
10
12

$ java DoWhileLoops
Enter an integer: -9
-8
-6
-4
-2
0
```

Sample Solution

```
$ java DoWhileLoops
Enter an integer: 12
0
2
4
6
8
10
12

$ java DoWhileLoops
Enter an integer: -9
-8
-6
-4
-2
0
```

```java
import java.util.Scanner;

public class DoWhileLoops {
    public static void main (String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter an integer: ");
        int maxVal = input.nextInt();

        if (maxVal >= 0) {
            int val = 0;
            do {
                System.out.println(val);
                val +=2;
            } while (val <= maxVal);
        } else {
            int val = maxVal - maxVal % 2;
            do {
                System.out.println(val);
                val +=2;
            } while (val <=0);
        }
    }
}
```