CS1101 Programming and Problem Solving

Dr. Gina Bai

Spring 2023

Logistics

- **ZY-3** on **zyBook > Assignments**
  - Due: Saturday, Feb 4, at 11:59pm
- **PA03 – W, A, B** on **zyBook > Chap 11**
  - Due: **Saturday, Feb 11**, at 11:59pm
  - Try to complete it before Exam 1

Recitation Sessions Starting Next Week

- Recitations are NOT lectures / office hours
  - Smaller group
  - Recap the materials covered in most recent lectures
    - No new topics/concepts
  - Practice on examples with peers

- Tue & Wed, 4:15pm – 5:15pm, 5:30pm – 6:30pm
- Stevenson 5306

Recitation Structure

- (~10min) Recitation leaders will briefly summarize and discuss
  - The common **mistakes/deductions** observed in previous PA
  - The recent **FAQs** during office hours
- (10~15min) **Q&A** Time
  - The assignments (PAs and ZYs), lecture materials, the practice exams, the returned midterm exams, and anything else
- (25~30min) Work on **exercises**
  - Quizzes and/or short coding exercises
  - Exam review problems

Recap

1. Parentheses: ()
2. Unary operators: +, -, !
3. Multiplicative operators: *, /, %
4. Additive operators: +, -
5. Relational operators: <, >, <=, >=
6. Equality operators: ==, !=
7. Logical AND: &&
8. Logical OR: ||
9. Assignment operators: =, +=, -=, *=, /=, %=

```
int x = 10;
int y = 5;
int z = 12;
System.out.println( x <= y );    // false
System.out.println( y < x && y <= z );    // true
System.out.println( x / y + x == z && z > 20 );    // false
System.out.println( x <= 2 * y && x >= 2 * y && z > 4 );    // true
System.out.println( !(x < y && x < z) );    // true
```

Conditionals

zyBook Chap 4.1 - 4.7

Why Conditionals?

Conditionals allow us to instruct the computer to **execute different lines of code** depending on whether **a condition** is **true or false**.

- Examples:
  - Acceptance into grad school based on undergrad GPA
  - Converting numerical grade to a letter grade
  - BMI ranges
  - Age restrictions

Conditional Structures

- `if` statements
  - Sequential `if` statements
- `if-else` statements
  - Nested `if-else` statements

if Statement

- Case 1: <condition> is `true`
  - **Execute the <controlled statement(s)>** within the { }
  - Continue to **execute the <statement(s)>**

- Case 2: <condition> is `false`
  - Skip the <controlled statement(s)>
  - **Execute the <statement(s)>**

```
if (<condition>) {
    <controlled statement(s)>;
}
<statement(s)>;
```

if Statement – Code Example

Write a program that accepts applications to graduate school **if** the student's GPA is greater than or equal to 3.0.

```
import java.util.Scanner;

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

        Scanner input = new Scanner(System.in);
        System.out.print("Enter GPA: ");
        double gpa = input.nextDouble();

        if(gpa >= 3.0) {
            System.out.println("Application accepted.");
        }
        System.out.println("Thank you for applying!");
    }
}
```

```
$ javac GradAdmission.java
$ java GradAdmission
Enter GPA: 3.2
Application accepted.
Thank you for applying!

$ java GradAdmission
Enter GPA: 2.99
Thank you for applying!
```

Q: What's the exact output of the following code?

```
int x = 12;

if (x >= 12) {
    System.out.print("A");
}
System.out.print("B");
```

AB

Sequential if Statement

- A **sequence** of if statements that would be evaluated (and maybe executed based on the truth value of the condition) **one by one**
  - Each **condition** and **its associated actions** is **independent** to others.

```
if (<cond1>) {
    <cond1 controlled statement(s)>;
}
if (<cond2>) {
    <cond2 controlled statement(s)>;
}
if (<cond3>) {
    <cond3 controlled statement(s)>;
}
<statement(s)>;
```

if-else Statement

- Case 1: <condition> is **`true`**
  - Execute the **<if controlled statement(s)>**
  - Continue to execute the **<statement(s)>**

- Case 2: <condition> is **`false`**
  - Execute the **<else controlled statement(s)>**
  - Continue to execute the **<statement(s)>**

```
if (<condition>) {
    <if controlled statement(s)>;
} else {
    <else controlled statement(s)>;
}
<statement(s)>;
```

if-else Statement – Code Example

Write a program that accepts applications to graduate school **if** the student's GPA is greater than or equal to 3.0. **Otherwise**, ask for an additional essay.

```
import java.util.Scanner;

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

        Scanner input = new Scanner(System.in);
        System.out.print("Enter GPA: ");
        double gpa = input.nextDouble();

        if(gpa >= 3.0) {
            System.out.println("Application accepted.");
        } else {
            System.out.println("Please submit an essay.");
        }
        System.out.println("Thank you for applying!");
    }
}
```

```
$ javac GradAdmission.java
$ java GradAdmission
Enter GPA: 3.2
Application accepted.
Thank you for applying!

$ java GradAdmission
Enter GPA: 2.99
Please submit an essay.
Thank you for applying!
```

Q: What's the exact output of the following code?

```
int x = 12;

if (x != 12) {
    System.out.print("A");
} else {
    System.out.print("B");
}
System.out.print("C");
```

BC

Q: What's the exact output of the following code?

```
int x = 20;

if (x < 15) {
    System.out.print("A");
} else {
    if (x > 17) {
        System.out.print("B");
    } else {
        System.out.print("C");
    }
}
System.out.print("D");
```

BD

Nested if-else Statement

```
int x = 20;

if (x < 15) {
    System.out.print("A");
} else {
    if (x > 17) {
        System.out.print("B");
    } else {
        System.out.print("C");
    }
}
System.out.print("D");
```

Equivalent Implementation

```
int x = 20;

if (x < 15) {
    System.out.print("A");
} else if (x > 17) {
        System.out.print("B");
} else {
        System.out.print("C");
}
System.out.print("D");
```

Nested if-else Statement – end with else

- Case 1: <cond1> is **true**
  - Execute the **<cond1 controlled stmt(s)>**
  - Continue to execute the **<statement(s)>**
- Case 2: <cond1> is **false** && <cond2> is **true**
  - Execute the **<cond2 controlled stmt(s)>**
  - Continue to execute the **<statement(s)>**
- Case 3: <cond1> is **false** && <cond2> is **false**
  - Execute the **<else controlled stmt(s)>**
  - Continue to execute the **<statement(s)>**

```
if (<cond1>) {
    <cond1 controlled stmt(s)>;
} else if (<cond2>) {
    <cond2 controlled stmt(s)>;
} else {
    <else controlled stmt(s)>;
}
<statement(s)>;
```

NOTE:
Exactly One of these sets of controlled statements will be executed

Nested if-else Statement – Example 1

Write a program that determines if an input integer is positive, negative, or zero.

```java
import java.util.Scanner;

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

        Scanner input = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        int number = input.nextInt();

        if(number > 0) {
            System.out.println("Positive");
        } else if(number < 0){
            System.out.println("Negative");
        } else {
            System.out.println("Zero");
        }
    }
}
```

```
$ javac NumberInfo.java
$ java NumberInfo
Enter an integer: 1
Positive

$ java NumberInfo
Enter an integer: -1
Negative

$ java NumberInfo
Enter an integer: 0
Zero
```

Nested if-else Statement – end with else if

- Case 1: <cond1> is **true**
  - Execute the **<cond1 controlled stmt(s)>**
  - Continue to execute the **<statement(s)>**
- Case 2: <cond1> is **false** && <cond2> is **true**
  - Execute the **<cond2 controlled stmt(s)>**
  - Continue to execute the **<statement(s)>**
- Case 3: <cond1> is **false** && <cond2> is **false**
  - Continue to execute the **<statement(s)>**

```
if (<cond1>) {
    <cond1 controlled stmt(s)>;
} else if (<cond2>) {
    <cond2 controlled stmt(s)>;
}
<statement(s)>;
```

NOTE:
Zero or One of these sets of controlled statements will be executed

Nested if-else Statement – Example 2

Write a program that determines if the participant earns the first place, the second place, or the third place.

```java
import java.util.Scanner;

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

        Scanner input = new Scanner(System.in);
        System.out.print("Enter place (int): ");
        int place = input.nextInt();

        if(place == 1) {
            System.out.println("First");
        } else if(place == 2){
            System.out.println("Second");
        } else if(place == 3){
            System.out.println("Third");
        }
        System.out.println("Thank you for running the race!");
    }
}
```

```
$ javac RaceResult.java
$ java RaceResult
Enter place (int): 1
First
Thank you for running the race!

$ java RaceResult
Enter place (int): 2
Second
Thank you for running the race!

$ java RaceResult
RaceResult
Enter place (int): 3
Third
Thank you for running the race!

$ java RaceResult
Enter place (int): 4
Thank you for running the race!
```

Coding Practice

- There is a program called LetterGrade, which prompts the user for the numerical grade (in whole number), and converts it to letter grade. It behaves like…

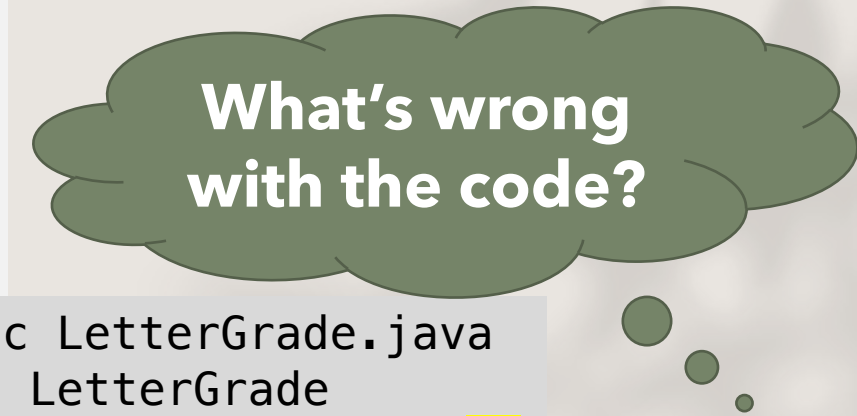
```java
import java.util.Scanner;

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

        Scanner input = new Scanner(System.in);
        System.out.print("Grade as whole number: ");
        int grade = input.nextInt();

        char letter = ' ';

        if(grade >= 90) {
            letter = 'A';
        }
        if(grade >= 80) {
            letter = 'B';
        }
        if(grade >= 70) {
            letter = 'C';
        }
        if(grade >= 60) {
            letter = 'D';
        }
        if(grade < 60) {
            letter = 'F';
        }
        System.out.println("The letter grade is: " + letter);
    }
}
```




```
$ javac LetterGrade.java
$ java LetterGrade
Grade as whole number: 99
The letter grade is: D

$ java LetterGrade
Grade as whole number: 83
The letter grade is: D

$ java LetterGrade
Grade as whole number: 47
The letter grade is: F

$ java LetterGrade
Grade as whole number: 60
The letter grade is: D

$ java LetterGrade
Grade as whole number: 74
The letter grade is: D
```

Sample Solution 1

```java
import java.util.Scanner;

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

        Scanner input = new Scanner(System.in);
        System.out.print("Grade as whole number: ");
        int grade = input.nextInt();

        char letter = ' ';

        if(grade >= 90) {
            letter = 'A';
        } else if(grade >= 80) {
            letter = 'B';
        } else if(grade >= 70) {
            letter = 'C';
        } else if(grade >= 60) {
            letter = 'D';
        } else {
            letter = 'F';
        }
        System.out.println("The letter grade is: " + letter);
    }
}
```

Nested if-else ends with else

```
$ javac LetterGrade.java
$ java LetterGrade
Grade as whole number: 99
The letter grade is: A

$ java LetterGrade
Grade as whole number: 83
The letter grade is: B

$ java LetterGrade
Grade as whole number: 47
The letter grade is: F

$ java LetterGrade
Grade as whole number: 60
The letter grade is: D

$ java LetterGrade
Grade as whole number: 74
The letter grade is: C
```

Sample Solution 2

```java
import java.util.Scanner;

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

        Scanner input = new Scanner(System.in);
        System.out.print("Grade as whole number: ");
        int grade = input.nextInt();

        char letter = ' ';

        if(grade >= 90) {
            letter = 'A';
        } else if(grade >= 80) {
            letter = 'B';
        } else if(grade >= 70) {
            letter = 'C';
        } else if(grade >= 60) {
            letter = 'D';
        } else if(grade < 60) {
            letter = 'F';
        }
        System.out.println("The letter grade is: " + letter);
    }
}
```

Nested if-else ends with else if

```
$ javac LetterGrade.java
$ java LetterGrade
Grade as whole number: 99
The letter grade is: A

$ java LetterGrade
Grade as whole number: 83
The letter grade is: B

$ java LetterGrade
Grade as whole number: 47
The letter grade is: F

$ java LetterGrade
Grade as whole number: 60
The letter grade is: D

$ java LetterGrade
Grade as whole number: 74
The letter grade is: C
```

Sample Solution 3

```java
import java.util.Scanner;

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

        Scanner input = new Scanner(System.in);
        System.out.print("Grade as whole number: ");
        int grade = input.nextInt();

        char letter = ' ';

        if(grade >= 90) {
            letter = 'A';
        }
        if(grade >= 80 && grade < 90) {
            letter = 'B';
        }
        if(grade >= 70 && grade < 80) {
            letter = 'C';
        }
        if(grade >= 60 && grade < 70) {
            letter = 'D';
        }
        if(grade < 60) {
            letter = 'F';
        }
        System.out.println("The letter grade is: " + letter);
    }
}
```

Specify Ranges

```
$ javac LetterGrade.java
$ java LetterGrade
Grade as whole number: 99
The letter grade is: A

$ java LetterGrade
Grade as whole number: 83
The letter grade is: B

$ java LetterGrade
Grade as whole number: 47
The letter grade is: F

$ java LetterGrade
Grade as whole number: 60
The letter grade is: D

$ java LetterGrade
Grade as whole number: 74
The letter grade is: C
```

Sample Solution 4

```java
import java.util.Scanner;

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

        Scanner input = new Scanner(System.in);
        System.out.print("Grade as whole number: ");
        int grade = input.nextInt();

        char letter = ' ';

        if(grade < 60) {
            letter = 'F';
        }
        if(grade >= 60) {
            letter = 'D';
        }
        if(grade >= 70) {
            letter = 'C';
        }
        if(grade >= 80) {
            letter = 'B';
        }
        if(grade >= 90) {
            letter = 'A';
        }
        System.out.println("The letter grade is: " + letter);
    }
}
```

Flip the order of the sequential if statements

```
$ javac LetterGrade.java
$ java LetterGrade
Grade as whole number: 99
The letter grade is: A

$ java LetterGrade
Grade as whole number: 83
The letter grade is: B

$ java LetterGrade
Grade as whole number: 47
The letter grade is: F

$ java LetterGrade
Grade as whole number: 60
The letter grade is: D

$ java LetterGrade
Grade as whole number: 74
The letter grade is: C
```