

CS1101

Programming and Problem Solving

Dr. Gina Bai
Spring 2023

Logistics

- **PA02 - W1, W2, A, B** on **zyBook > Chap 11**
 - Due: Thursday, Feb 2, at 11:59pm
 - NOTE: Replace the header comment for PA02 - A
- **ZY-3** on **zyBook > Assignments**
 - Due: Saturday, Feb 4, at 11:59pm

Midterm Exam 1 – Format

- All materials correspond to zyBook **Chapters 1 - 3** and **part of Chapter 4**
- **Paper-based**, closed book, closed notes
 - Tip: practice writing code on paper
- A combination of short answer, multiple choice, true/false, code reading and writing
- Regular class time (50 minutes)
 - Arrive early!!!

Midterm Exam 1 – Additional Support

Reference guide

- We will provide a complete program as a *reference guide* in your exam packet and is meant to help you with syntax questions
 - Includes methods that receive parameters, methods that return parameters, user input statements, print statements, and decision statements

Midterm Exam 1 – Programming Style

- You *do not* need to comment your code
- You *do not* need to keep track of line length
- You *do not* need to use meaningful names for identifiers
- You ***NEED*** to use **proper indentation**
- Your answer must be ***LEGIBLE***

Learning Objectives – Chap 1

- General computer science terminology
- Java terminology (e.g., compile, bytecode, JVM, interpreter, ...)
- Java program structure (e.g., class, methods, statements, ...)
- Using a `Scanner` object to get input from the user
 - **Prompt for user input first**, and then read in input with `Scanner`
- Output statements using `print`, `println`, and `printf`
- Types of programming errors: syntax, runtime, and logic errors
 - Identify, and describe if the program would compile and run

Learning Objectives – Chap 2

- Java identifiers (e.g., naming convention of Class/method/variable)
- Declaring, initializing/assigning, and using variables
- Various data types
- Expressions
- Arithmetic operators and operator precedence
- **Type conversion:** promotion/coercion and type casting
- **Math** class methods
- **Escape sequence**

Learning Objectives – Chap 3

- Problem decomposition
- Declaring methods and calling methods
 - Return type, accepts parameters or not, data type of the parameters
- Using parameters to pass information to a method
 - Pass-by-value vs. pass-by-reference
- Variable scope
- Returning a value from a method

Learning Objectives – Chap 4

- Decision statement structure
- Equality, Relational and Logic operators, and operator precedence

Overview – Code Reading

Reading segments of Java code

- Determining the output they produce
- Determining the value of variables
- Finding errors
- Tracing through control flow constructs
- Tracing method calls and passed parameters
- When asked to show the output that a piece of code, *be precise* (e.g., do not list the output on a single line if it would appear on multiple lines)

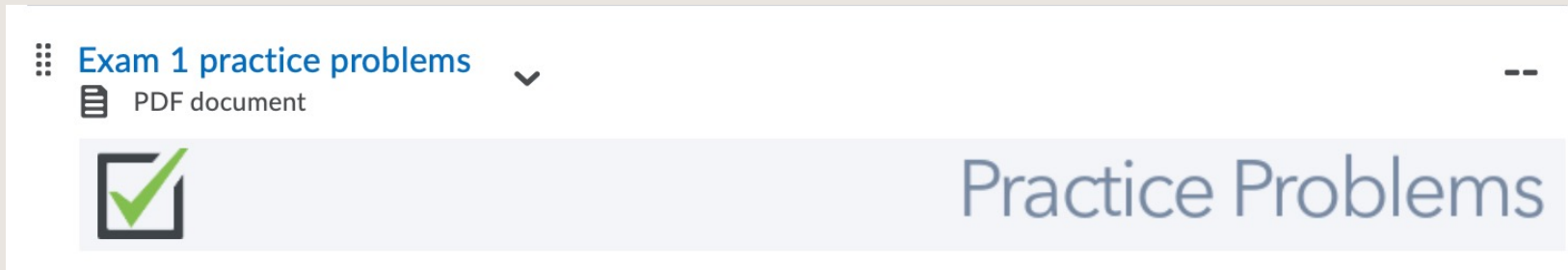
Midterm Exam 1 – Preparation Strategies

- Review the learning objectives
- Review the lecture slides
- Review zyBook, and the activities
- Review (and possibly rewrite) the lab exercises
- If you do not fully understand a topic, read the related textbook section
- Attend office hours to ask additional questions/clarifications
- Complete the Practice Exam

Midterm Exam 1 – Practice Exam

Practice Exam

- **Brightspace > Content > Course Documents**
- Solutions will be posted by Sunday, Feb 5th
- *Will not* demonstrate *all* the kinds of problems you can see
- Not a full view of the exam, *just a snapshot*



Boolean Data Type

zyBook Chap 4.12

Boolean

- A data type whose values are **true** and **false**
- Boolean expressions are commonly used for **logical tests**
- It is legal to
 - create a boolean **variable**
 - pass a boolean value as a **parameter**
 - **return** a boolean value from methods
 - call a method that returns a boolean and use it as a test

Equality & Relational & Logical Operators

zyBook Chap 4.2, 4.4, 4.5, 4.10

Equality and Relational Operators

- Compare two expressions
- Result in boolean (true or false)
- Only use with **primitive data**

Operator	Description	Example	Result
<code>==</code>	Equal to	<code>2 + 2 == 4</code>	true
<code>!=</code>	Not equal to	<code>3.2 != 4.1</code>	true
<code><</code>	Less than	<code>4 < 3</code>	false
<code>></code>	Greater than	<code>4 > 3</code>	true
<code><=</code>	Less than or equal to	<code>2 <= 0</code>	false
<code>>=</code>	Greater than or equal to	<code>2.5 >= 2.5</code>	true

Logical Operators

- Conditions can be combined with logical operators
- We use truth tables to evaluate logical operators.

Operator	Description	Example	Result
&&	AND	(2 == 3) && (-1 < 5) false && true	false
	OR	(2 == 3) (-1 < 5) false true	true
!	NOT	!(2 == 3) !(false)	true

p	q	!p	p && q	p q
true	true	false	true	true
true	false	false	false	true
false	true	true	false	true
false	false	true	false	false

“Exclusive OR” vs. “Inclusive OR”

- OR in natural language
 - Exclusive OR
 - A OR B
 - Case 1: A is true
 - Case 2: B is true
- **OR in programming language**
 - **Inclusive OR**
 - **A OR B**
 - Case 1: A is true
 - Case 2: B is true
 - Case 3: Both A and B are true

Precedence

If two operations are at the same precedence order, evaluate from left to right with the exception of assignment operators that are evaluated right to left.

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: =, +=, -=, *=, /=, %=

Evaluate → Compare/Combine → Assign

Q: Determine the truth value of

false || true && -5 / 2 + (13 + 6) < 19

false || true && -5 / 2 + 19 < 19

false || true && -2 + 19 < 19

false || true && 17 < 19

false || true && true

false || true

true

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: =, +=, -=, *=, /=, %=

Short Circuit Evaluation

zyBook Chap 4.18

Short Circuit Evaluation

Java stops evaluating a test if it knows the answer.

- `&&` stops early if any part of the test is `false`
- `||` stops early if any part of the test is `true`

```
2 > 3 && 4 < 5
false && <skipped>
false
```

```
2 < 3 || 4 > 5
true || <skipped>
true
```


Q: If `year` is 2023 and `month` is 2, which of the following expressions will short circuit?

- `year > 2000 && month > 6` ✗
- `year > 2000 || month < 12` ✓
- `year < 2000 || month < 12` ✗
- `year < 2000 && month > 6` ✓

De Morgan's Laws

De Morgan's Laws

- Rules used to **negate** boolean tests.
- Useful when you want the **opposite** of an existing test.

Original Expression	Negated Expression	Simplified Negated Expression
<code>a && b</code>	<code>!(a && b)</code>	<code>!a !b</code>
<code>a b</code>	<code>!(a b)</code>	<code>!a && !b</code>

→ Negate the whole thing
→ Flip the logical operator
→ Distribute the negation

Q: Negate $(x > y) \ \&\& \ (y > z)$

$!((x > y) \ \&\& \ (y > z))$

$!(x > y) \ || \ !(y > z)$

$x \leq y \ || \ y \leq z$

Q: Negate $(x == y) \ || \ (x \leq z)$

$!((x == y) \ || \ (x \leq z))$

$!(x == y) \ \&\& \ !(x \leq z)$

$x \neq y \ \&\& \ x > z$