

# Midterm Study Guide

[ Note: These are rough numbers ]

## Midterm question breakdown

Subject	number of questions (Approx)
About computers, java compiler, primitive data types, and variables	10
What is the value of X or result of Expression	25
Object methods (String, Scanner, PrintWriter, File, Random)	6
Comparison operators	3
Files	6
<b>TOTAL</b>	<b>50</b>

Within the "What is the result of the expression" type questions, these subjects are included:

Subject	Number of questions (Approx)
Arithmetic operators	5
if / else if / else	4
Logical operators	1
Switch	2
printf	1
loops (while, do while, for)	8
Increment Operator	2
Scanner	2
<b>TOTAL</b>	<b>25</b>

## What do I study?

Read the Chapter Lecture Notes (Chapters 2 - 4). While you read the notes, practice creating and running the file examples in IntelliJ. Edit the files and re-run them to see how they change.

Also study the slides and read the course Source Code for chapters 2-4 in IntelliJ.

Learning to Read and Write Java is the most important thing. The midterm is mostly testing you on how to READ Java. This means that you understand what the output of a variable or an expression will be. Studying means practicing these evaluations using IntelliJ. On the exam, you have to demonstrate you know these things.

## What is the value of X or result of this expression?

Understand what the results of these expressions are and more importantly why (answer by yourself at first, then validate using IntelliJ or JShell):

```
7 % 5;  
  
5 + 2 * 10 - 3;  
  
int i = 33;  
i / 10;  
  
double i = 33.0;  
i / 10;
```

What is the value of x (try to answer without programming it. If you can't, run the code and tweak the code, until you understand it).

```
// First  
int x = 5;  
int y = 10;  
if (x < y) {  
    x = y  
}  
System.out.println(x)  
  
// Second  
x = 50;  
y = 10;  
if (x != y)  
    x -= y;  
System.out.println(x)  
  
// Third  
x = 50;  
y = 50;  
x++;
```

```

if (x < y){
    x = 500;
}
else if (x > y) {
    x = 0;
}
else {
    x = 1000;
}
System.out.println(x)

// Fourth
double x;
char letter = 'Y';
switch (letter) {
    case 'A':
        x = 0.3;
        break;
    case 'B':
        x = 0.5;
        break;
    default:
        x = 0.7;
}
System.out.println(x)

// Fifth
char someLetter = 'B';
double x = switch(someLetter){
    case 'A' -> 0.369;
    case 'B' -> 0.571;
    default -> 0.792;
};
System.out.printf("%.2f %n",x);

// Sixth
int x = 0;
while (x < 11){
    x += 5;
}
System.out.println(x)

// Seventh -- what is x and what is y
int x = 5;
int y = 7;
x = ++y;

```

```

System.output.println(x)
System.output.println(y)
x = y++;
System.output.println(x)
System.output.println(y)

// Eighth
int x = 0;
for (int y = 20; y < 50; y+=2) {
    x = y
}
System.output.println(x)

// Ninth
for (int number = 1; number < 10; number+=3) {
    System.out.print(number + ", ");
}

// Tenth – How many times will this loop execute?
int x = 5;
int y = 10;
while (y < 20) {
    x++
    System.output.println(x)
}

```

## File Objects and Methods

Here is an example of everything you need to know about file methods and objects:

```

import java.io.*;
import java.util.Scanner;

public class AboutFilesEtc {
    public static void main(String[] args) throws IOException {
        File myFile = new File("filename.txt");
        FileWriter fw = new FileWriter(myFile, true);
        PrintWriter outputFile = new PrintWriter(fw);
        outputFile.println("This is LINE ONE");
        outputFile.print("This is LINE TWO \n");
        outputFile.printf("This is %s %d %n", "LINE", 3);
        outputFile.println("This is THE FINAL LINE");
        outputFile.close();

        Scanner inputFile = new Scanner(myFile);
    }
}

```

```

        while(inputFile.hasNext()) {
            String str = inputFile.nextLine();
            System.out.println(str);
        }
    }
}

```

Here's the same file with comments:

```

import java.io.*;
import java.util.Scanner;

public class AboutFilesEtc {
    public static void main(String[] args) throws IOException {
        // Instantiate a File object that references "filename.txt"
        // This file either exists, or will be created.
        File myFile = new File("filename.txt");

        // Instantiate a FileWriter object to append data to the file.
        FileWriter fw = new FileWriter(myFile, true);

        // Instantiate a PrintWriter object to print data to the file.
        // Then use the print() methods to write the data.
        // close() the file to save the data.
        PrintWriter outputFile = new PrintWriter(fw);
        outputFile.println("This is LINE ONE");
        outputFile.print("This is LINE TWO \n");
        outputFile.printf("This is %s %d %n", "LINE", 3);
        outputFile.println("This is THE FINAL LINE");
        outputFile.close();

        // Instantiate a Scanner object with a reference to the myFile
        object Scanner inputFile = new Scanner(myFile);

        // use the hasNext() method to loop through every line in the
        ifle.

        while(inputFile.hasNext()) {
            // Load the current line into a String variable.
            String str = inputFile.nextLine();
            // Print the str, or use string methods on the string.
            System.out.println(str);
        }
    }
}

```

# General

Understand:

- What Final does to a variable.
- The syntax of creating an object.
- The object methods we covered for String, Scanner, PrintWriter, File, and Random.
- How comparison differs between Primitive Data Types and Strings.
- New Switch.