

CS110T: Programming Language1

Lab 3: Java basics II



Lab Objectives:

In this lab, the student will practice:

- ✓ Defining variables of different data types
- ✓ Using format specifiers in printing output
- ✓ Compiling and executing Java applications.
- ✓ Writing and evaluating mathematical expressions

Lab Exercise 1: Program Output

Problem Description: What is the output of the following code?

```
// Lab Exercise 1: Math.java
// Computing and displaying the result of different expressions public
class Math
{
    // main method begins execution of Java application      public static
void main( String args[] )
    {
        System.out.println("8" + 6 + 6);
        System.out.println(6 + "6" + 6);
        System.out.println(6+ 6 + "6");

        System.out.println("4.5+1="+ (4.5+1));
        System.out.println("4.0+9="+4.0+9);
        System.out.print("((2-2) × 1×3) - 3/1.0=-3.0");
        System.out.println(((2-2)*1*3)-3/1.0);
    }
}
```

866 // "8" + 6 + 6 = string concatenation

666 // 6 + "6" + 6 = string concatenation

126 // 6 + 6 + "6" = 12 + "6" = "126"

4.5+1=5.5 // "4.5+1=" + (4.5+1) = string + calculation

4.0+9=4.09 // "4.0+9=" + 4.0 + 9 = string concatenation

((2-2) × 1×3) - 3/1.0=-3.0-3.0 // Text + calculation result (-3.0)

Lab Exercise 2: Java statements (code writing 1)

A) Problem Description: Write Java statements that accomplish the following:

1. Create a variable named *studentName* and assign the value “Sara Khaled” to it.
2. Create a variable named *examMark* and assign the value 14.5 to it.
3. Declare a constant named *HOMEWORK* and assign value 5 to it.
4. Calculate the total marks after adding the homework mark to the exam mark and save it in a new variable.
5. By using *printf* method and *escape characters*, Display the student Name, her exam mark and total mark as shown:

```

Student Name:      Exam Mark:      Total Mark :
Sara Khaled       14.5           19.5
BUILD SUCCESSFUL (total time: 0 seconds)

```

```

public class StudentMarks {
    public static void main(String[] args) {
// 1. Create studentName variable
        String studentName = "Sara Khaled";

// 2. Create examMark variable
        double examMark = 14.5;

// 3. Declare constant HOMEWORK
        final int HOMEWORK = 5;

// 4. Calculate total marks
        double totalMark = examMark + HOMEWORK;

// 5. Display using printf
        System.out.printf("Student Name:  Exam Mark:  Total Mark :%n");
        System.out.printf("%-16s %-14.1f %.1f:%.0f%n", studentName, examMark,
totalMark, (double)HOMEWORK);
    }
}

```

B) Problem Description: Write Java statements that accomplish the following:

6. Declare a constant named *DISCOUNT* and assign value 100 to it.
7. Create a variable named *productName* and assign the value “Tote Bag” to it.
8. Create a variable named *total* and assign the value 1499.95 to it.
9. Calculate the total after subtracting the discount amount and save it in a new variable.
10. By using *printf* method and *escape characters*, Display the product Name and its total before and after discount as shown:

```
Product Name:      Total :      Total after discount :
Tote Bag          1499.95      1399.95
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
public class ProductDiscount { public static void main(String[] args)
{ // 6. Declare constant DISCOUNT final double DISCOUNT
= 100;

// 7. Create productName variable
String productName = "Tote Bag";

// 8. Create total variable
double total = 1499.95;

// 9. Calculate total after discount double
totalAfterDiscount = total - DISCOUNT;

// 10. Display using printf
System.out.printf("%-15s %-15s %-20s%n", "Product Name:", "Total :", "Total
after discount");
System.out.printf("%-15s %-15.2f %-20.2f%n", productName, total,
totalAfterDiscount);
}
}
```

C) Problem Description: Write Java statements that accomplish the following:

1. Declare a constant named *DISCOUNT* and assign value 0.10 to it.
2. Create a variable named *customerName* and assign the value “Ahmed” to it.
3. Create a variable named *total* and assign the value 500 to it.
4. Calculate the total after discount 10% and save it in a new variable.
5. By using *printf* method and *escape characters*, Display the customer Name and his total before and after discount as shown:

```
Customer Name:      Total :      Total after discount :
Ahmed              500          450.00
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
public class CustomerDiscount {
    public static void main(String[] args) {
// 1. Declare constant DISCOUNT
final double DISCOUNT = 0.10;

// 2. Create customerName variable
String customerName = "Ahmed";

// 3. Create total variable
double total = 500;

// 4. Calculate total after discount      double
totalAfterDiscount = total - (total * DISCOUNT);

// 5. Display using printf
System.out.printf("%-15s %-15s %-25s%n", "Customer Name:", "Total :", "Total
after discount :");
System.out.printf("%-15s %-15.0f %-25.2f%n", customerName, total,
totalAfterDiscount);
    }
}
```

Lab Exercise 3: Java statements (code writing 2)

Write a java program that computes the calories burned in the exercise:

1. Declare a constant calories-per-minute rate (e.g., 8 calories per minute).
2. Store the workout duration in minutes as a double.
3. Calculate total calories burned and store it as an int.
4. Print workout duration (with 2 decimals) and calories (whole number). Using `System.out.printf()`

Hint: total calories = calories per minute * workout duration

Output:

```
Workout Duration: 45.75 minutes  
Calories Burned: 366
```

```
public class CaloriesCalculator {  
public static void main(String[] args) {  
    // 1. Declare constant calories-per-minute rate  
final int CALORIES_PER_MINUTE = 8;  
  
    // 2. Store workout duration as double  
double workoutDuration = 45.75;  
  
    // 3. Calculate total calories burned as int    int totalCalories = (int)  
(CALORIES_PER_MINUTE * workoutDuration);  
  
    // 4. Print using printf  
    System.out.printf("Workout Duration: %.2f minutes%n", workoutDuration);  
    System.out.printf("Calories Burned: %d%n", totalCalories);  
}  
}
```

Lab Exercise 4: Code writing (4)

Problem description: You are creating a simple program for a library to generate library cards for new members. The library card should display the member's full name, a unique ID number, and a welcome message. Write a Java program that:

1. Takes the member's first name, last name, and a unique ID number (an integer) as input.
2. Concatenates the first name and last name to form the full name.
3. Constructs a welcome message using string concatenation that includes the member's full name and their unique ID.
4. Prints the full name and the welcome message using `System.out.println()`.

Example Scenario: (You can use your own name)

- The member's first name is "Fatimah".
- The member's last name is "Ahmed".
- The unique ID is 12678.

Requirements:

- Use string concatenation to combine the first name and last name.
- Use string concatenation to create the welcome message.
- Use `System.out.println()` to display the results.

Output:

```
Full name: Fatimah Ahmed  
Welcome message: Welcome, Fatimah Ahmed! Your library ID is 12345.
```

```
import java.util.Scanner;  
  
public class LibraryCard {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        // 1. Take input from user  
        System.out.print("Enter first name: ");  
        String firstName = scanner.nextLine();  
  
        System.out.print("Enter last name: ");  
        String lastName = scanner.nextLine();  
  
        System.out.print("Enter unique ID: ");  
        int uniqueID = scanner.nextInt();  
  
        // 2. Concatenate first and last name  
        String fullName = firstName + " " + lastName;  
  
        // 3. Construct welcome message  
        String welcomeMessage = "Welcome, " + fullName + "! Your library ID is " +  
uniqueID + ".";  
  
        // 4. Print results  
        System.out.println("Full name: " + fullName);  
        System.out.println("Welcome message: " + welcomeMessage);  
  
        scanner.close();  
    }  
}
```

Lab3 Assignment Problems

Problem Description1: You are developing a weather app that provides temperature updates. The app tracks the current temperature in Fahrenheit (a double) and needs to convert it to Celsius (an integer). Write a Java program that:

1. Takes the current temperature in Fahrenheit as input.
2. Converts the temperature to Celsius using the formula $C = (F - 32) * 5/9$.
3. Casts the Celsius temperature to an integer.

- Prints the original temperature in Fahrenheit, the converted temperature in Celsius (before casting), and the casted temperature (as an integer) using `System.out.println()` statements.

Example Scenario:

- The current temperature is 77.5°F.

Output

```
Original temperature in Fahrenheit: 77.5
Converted temperature in Celsius: 25.27777777777778   Celsius
temperature rounded to whole number: 25
```

```
import java.util.Scanner;
```

```
public class TemperatureConverter {
```

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

```
    Scanner scanner = new Scanner(System.in);
```

```
    // 1. Take temperature input
```

```
    System.out.print("Enter temperature in Fahrenheit: ");
```

```
double fahrenheit = scanner.nextDouble();
```

```
    // 2. Convert to Celsius    double
```

```
celsius = (fahrenheit - 32) * 5/9;
```

```
    // 3. Cast to integer
```

```
int celsiusInt = (int) celsius;
```

```
    // 4. Print results
```

```
    System.out.println("Original temperature in Fahrenheit: " + fahrenheit);
```

```
    System.out.println("Converted temperature in Celsius: " + celsius);
```

```
    System.out.println("Celsius temperature rounded to whole number: " + celsiusInt);
```

```
    scanner.close();
```

```
}
```

```
}
```

Problem Description2: Consider the following Java code and predict the output for each of the casting operations:

1. What will be the output of casting the double value 9.78 to an int?
2. What will be the output of casting the int value 100 to a byte?
3. What will be the output of casting the long value 150L to a float?
4. What will be the output of casting the char value 'A' to an int?

```
public class CastingExample {      public
static void main(String[] args) {
double dValue = 9.78;          int iValue =
(int) dValue;

    int num = 100;
    byte bValue = (byte) num;

    long lValue = 150L;
    float fValue = (float) lValue;

    char cValue = 'A';
int cToInt = (int) cValue;

    System.out.println("Original double value: " +
dValue);          System.out.println("Casted to int: " +
iValue);
    System.out.println("Original int value: " +
num);          System.out.println("Casted to byte: " +
bValue);
    System.out.println("Original long value: " + lValue);
System.out.println("Casted to float: " + fValue);
    System.out.println("Original char value: " + cValue);
    System.out.println("Casted to int: " + cToInt);
}
}
```

Original double value: 9.78

Casted to int: 9

Original int value: 100

Casted to byte: 100

Original long value: 150

Casted to float: 150.0

Original char value: A

Casted to int: 65

Problem Description3: Complete the following code and write the output of code:

```
// Lab Exercise 1: Math.java
// Computing and displaying the result of different expressions public class
Math
{
    // main method begins execution of Java application public
    static void main( String args[] )
    {
        myNum =
        myFloatNum =
        8.99myLetter = 'A';
        myBool = false;    myText
        World";
        System.out.println(myText);
        System.out.println("4.0+9="+4.0+myNum);
        System.out.print("((2-2) × 1×3) - 3/1.0= ");
        System.out.println(((2-2)*1*3)-3/1.0);
        System.out.println("(1+2)*3*4/5 =" + (1+2)*3*4/5);
    }
}
```

```
public class Math { public static void
main(String[] args) { // Complete the
variable declarations int myNum = 9;
float myFloatNum = 8.99f; char
myLetter = 'A'; boolean myBool =
false;
String myText = "Hello World";
```

```
System.out.println(myText); // Output:
```

Hello World

```
System.out.println("4.0+9="+4.0+myNum); // Output:
```

4.0+9=4.09 (string concatenation)

```
System.out.print("((2-2) × 1×3) - 3/1.0= "); // Output:
```

((2-2) × 1×3) - 3/1.0=

```
System.out.println(((2-2)*1*3)-3/1.0); // Output:
```

-3.0

```
System.out.println("(1+2)*3*4/5 =" + ((1+2)*3*4/5)); // Output:
```

(1+2)*3*4/5 =7 (integer division)

```
}
```

```
}
```

Hello World

4.0+9=4.09

((2-2) × 1×3) - 3/1.0= -3.0

(1+2)*3*4/5 =7

