Lab 03

Java Fundamentals-II

Objective:

This lab is designed to give you practice with some of the fundamental's concepts in Java.

Activity Outcomes:

On completion of this lab student will be able

- Demonstrate the concept of literals.
- Write expressions using various operators (arithmetic, increment/decrement).
- Demonstrate the type conversion used in program.
- Display formatted output on screen.

Instructor Note:

As a pre-lab activity, read Chapter 02 from the text book "Java How to Program, Deitel, P. & Deitel, H., Prentice Hall, 2019".

1) Useful Concepts

Literals: A literal is a constant value that appears directly in a program.

20	**Integer literal**
12L or 123l	Force integer literal to be treated as a long
"Today we sold "	String literal
12.3	Floating point literal
12.3f	Force a double literal to be treated as a float

Scientific and E notation

Decimal Notation	Scientific Notation	E Notation
247.91	$2.4791 \cdot 10^{2}$	2.4791E2
0.00072	$7.2 \cdot 10^{-4}$	7.2E–4
2,900,000	$2.9 \cdot 10^{6}$	2.9E6

Arithmetic Operators:

Operator	Meaning	Type	Example
+	Addition	Binary	`total = cost + tax;`
-	Subtraction	Binary	`cost = total - tax;`
*	Multiplication	Binary	`tax = cost * rate;`
/	Division	Binary	`salePrice = original / 2;`
%	Modulus	Binary	`remainder = value % 3;`

An **arithmetic expression** is constructed by using arithmetic operators and numbers.

Precedence of arithmetic operators (highest to lowest)

Highest Precedence →	- (unary negation)
	* / %
Lowest Precedence →	+ -

Associativity of arithmetic operators

Operator	Associativity
- (unary negation)	Right to left
* / %	Left to right
+ -	Left to right

Combined assignment operators

Operator	Example Usage	Equivalent To
`+=`	`x += 5;`	`x = x + 5;`
`-=`	`y -= 2;`	`y = y - 2;`
`*=`	`z *= 10;`	`z = z * 10;`
`/=`	`a /= b;`	`a = a / b;`
`%=`	`c %= 3;`	`c = c % 3;`

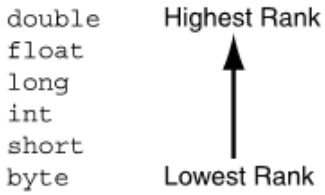
Increment and Decrement Operators

Operator	*Name*	*Description*	*Example (assume i = 1)*
`++var`	preincrement	Increment `var` by `1`, and use the new `var` value in the statement	`int j = ++i;` `// j is 2, i is 2`
`var++`	postincrement	Increment `var` by `1`, but use the original `var` value in the statement	`int j = i++;` `// j is 1, i is 2`
`--var`	predecrement	Decrement `var` by `1`, and use the new `var` value in the statement	`int j = --i;` `// j is 0, i is 0`
`var--`	postdecrement	Decrement `var` by `1`, and use the original `var` value in the statement	`int j = i--;` `// j is 1, i is 0`

Conversion between primitive Data Types

Java performs some conversions between data types automatically, but does not automatically perform any conversion that can result in the loss of data

Primitive data type ranking



Cast operators: The cast operator lets you manually convert a value

Statement	Description
`littleNum = (short)bigNum;`	The cast operator returns the value in `bigNum`, converted to a `short`. The converted value is assigned to the variable `littleNum`.
`x = (long)3.7;`	The cast operator is applied to the expression 3.7. The operator returns the value 3, which is assigned to the variable `x`.
`number = (int)72.567;`	The cast operator is applied to the expression 72.567. The operator returns 72, which is used to initialize the variable `number`.
`value = (float)x;`	The cast operator returns the value in `x`, converted to a `float`. The converted value is assigned to the variable `value`.
`value = (byte)number;`	The cast operator returns the value in `number`, converted to a `byte`. The converted value is assigned to the variable `value`.

Formatting Console Output

You can use the `System.out.printf` method to display formatted output on the console.

Format Specifier	*Output*	*Example*
`%b`	a Boolean value	true or false
`%c`	a character	'a'
`%d`	a decimal integer	200
`%f`	a floating-point number	45.460000
`%e`	a number in standard scientific notation	4.556000e+01
`%s`	a string	"Java is cool"

Example	*Output*
`%5c`	Output the character and add four spaces before the character item, because the width is 5.
`%6b`	Output the Boolean value and add one space before the false value and two spaces before the true value.
`%5d`	Output the integer item with width at least 5. If the number of digits in the item is < 5, add spaces before the number. If the number of digits in the item is > 5, the width is automatically increased.
`%10.2f`	Output the floating-point item with width at least 10 including a decimal point and two digits after the point. Thus, there are 7 digits allocated before the decimal point. If the number of digits before the decimal point in the item is < 7, add spaces before the number. If the number of digits before the decimal point in the item is > 7, the width is automatically increased.
`%10.2e`	Output the floating-point item with width at least 10 including a decimal point, two digits after the point and the exponent part. If the displayed number in scientific notation has width less than 10, add spaces before the number.
`%12s`	Output the string with width at least 12 characters. If the string item has fewer than 12 characters, add spaces before the string. If the string item has more than 12 characters, the width is automatically increased.

2) Solved Lab Activites

Sr.No	*Allocated Time*	*Level of Complexity*	*CLO Mapping*
Activity 1	***15 mins***	***Low***	***CLO-5***
Activity 2	***15 mins***	***Low***	***CLO-5***
Activity 3	***15 mins***	***Low***	***CLO-5***
Activity 4	***15 mins***	***Low***	***CLO-5***
Activity 5	***15 mins***	***Medium***	***CLO-5***

Activity-1:

This program illustrate the uses E notation

Solution:

```
public class Activity1{
      public static void main(String[] args){
            double distance, mass;
            distance = 1.495979E11;
            mass = 1.989E30;
            System.out.println("The sun is " + distance +" meters
      away.");
            System.out.println("The sun's mass is " + mass + "
      kilograms.");
      }
}
```

Output

```
The sun is 1.495979E11 meters away.
The sun's mass is 1.989E30 kilograms.
```

Activity-2:

This program calculates hourly wages plus overtime.

Solution:

```
public class Activity2{
      public static void main(String[] args){
            double regularWages;
            double basePay = 25;
            double regularHours = 40;
            double overtimeWages;
            double overtimePay = 37.5;
            double overtimeHours = 10;
```

```
            double totalWages;
            regularWages = basePay * regularHours;
            overtimeWages = overtimePay * overtimeHours;
            totalWages = regularWages + overtimeWages;
            System.out.println("Wages for this week are $" + totalWages);
      }
}
```

Output

```
Wages for this week are $1375.0
```

Activity-3:

This program calculates the amount of pay that will be contributed to a retirement plan if 5%, 8%, or 10% of monthly pay is withheld.

Solution:

```
public class Activity3{
    public static void main(String[] args){
       double monthlyPay = 6000.0;
       double contribution;
      // Calculate and display a 5% contribution.
       contribution = monthlyPay * 0.05;
       System.out.println("5 percent is $" + contribution +" per month.");
      // Calculate and display an 8% contribution.
      contribution = monthlyPay * 0.08;
      System.out.println("8 percent is $" + contribution + " per month.");
     // Calculate and display a 10% contribution.
      contribution = monthlyPay * 0.1;
      System.out.println("10 percent is $" + contribution + " per
month.");
   }
}
```

Output

```
5 percent is $300.0 per month.
8 percent is $480.0 per month.
10 percent is $600.0 per month.
```

Activity-4:

This program displays a variety of floating-point numbers in a column with their decimal points aligned.

Solution:

```
public class Activity4{
      public static void main(String[] args){
            // Declare a variety of double variables.
            double num1 = 127.899;
            double num2 = 3465.148;
            double num3 = 3.776;
            double num4 = 264.821;
            double num5 = 88.081;
            double num6 = 1799.999;
            // Display each variable in a field of
            // 8 spaces with 2 decimal places.
            System.out.printf("%8.2f\n", num1);
            System.out.printf("%8.2f\n", num2);
            System.out.printf("%8.2f\n", num3);
            System.out.printf("%8.2f\n", num4);
            System.out.printf("%8.2f\n", num5);
            System.out.printf("%8.2f\n", num6);
      }
}
```

Output

```
  127.90
 3465.15
    3.78
  264.82
   88.08
 1800.00
```

Activity-5:

This program displays the sales tax with two digits after the decimal point

Solution:

```
import java.util.Scanner;

public class Activity5 {

	public static void main(String[] args) {

		Scanner input = new Scanner(System.in);

		System.out.print("Enter purchase amount: ");

		double purchaseAmount = input.nextDouble();

		double tax = purchaseAmount * 0.06;

		System.out.println("Sales tax is $" + (int)(tax * 100) /
	100.0);

	}

}
```

Output

```
Enter purchase amount: 197.55
Sales tax is $11.85
```

3) Graded Lab Tasks

Note: The instructor can design graded lab activities according to the level of difficult and complexity of the solved lab activities. The lab tasks assigned by the instructor should be evaluated in the same lab.

Lab Task 1

Suppose you want to develop a program that changes a given amount of money into smaller monetary units. The program lets the user enter an amount as a double value representing a total in dollars and cents, and outputs a report listing the monetary equivalent in the maximum number of dollars, quarters, dimes, nickels, and pennies, in this order, to result in the minimum number of coins
Here are the steps in developing the program:

1. *Prompt the user to enter the amount as a decimal number, such as 11.56.*
2. *Convert the amount (e.g., 11.56) into cents (1156).*
3. *Divide the cents by 100 to find the number of dollars. Obtain the remaining cents using the cents remainder 100.*
4. *Divide the remaining cents by 25 to find the number of quarters. Obtain the remaining cents using the remaining cents remainder 25.*
5. *Divide the remaining cents by 10 to find the number of dimes. Obtain the remaining cents using the remaining cents remainder 10.*
6. *Divide the remaining cents by 5 to find the number of nickels. Obtain the remaining cents using the remaining cents remainder 5.*
7. *The remaining cents are the pennies.*
8. *Display the result.*

You are required to implement the above steps 1-8 in JAVA language

Lab Task 2

***N** students take **K** apples and distribute them among each other evenly. The remaining (the undivisible) part remains in the basket. How many apples will each single student get? How many apples will remain in the basket?*
*The program reads the numbers **N** and **K**. It should print the two answers for the questions above.*

Input:	***Output:***
6	*8*
50	*2*

Lab Task 3

A school decided to replace the desks in three classrooms. Each desk sits two students. Given the number of students in each class, print the smallest possible number of desks that can be purchased.
*The program should read three integers: the number of students in each of the three classes, **a, b** and **c** respectively.*

Input:	***Output:***
17	*28*
19	
18	

Lab Task 4

*Given the integer **N** – the number of minutes that is passed since midnight - how many hours and minutes are displayed on the 24h digital clock?*
The program should print two numbers: the number of hours (between 0 and 23) and the number of minutes (between 0 and 59).
*For example, if **N = 150**, then **150** minutes have passed since midnight - i.e. now is **2:30 am**. So the program should print **2 30**.*

Lab Task 5

A milk carton can hold 3.78 liters of milk. Each morning, a dairy farm ships cartons of milk to a local grocery store. The cost of producing one liter of milk is $0.38, and the profit of each carton of milk is $0.27. Write a program that does the following:

a. ***Prompts the user to enter the total amount of milk produced in the morning***
b. ***Outputs the number of milk cartons needed to hold milk (Round your answer to the nearest integer.)***
c. ***Outputs the cost of producing milk***
d. ***Outputs the profit for producing milk***

Lab Task 6

You found an exciting summer job for five weeks. It pays $15.50 per hour. Suppose that the total tax you pay on your summer job income is 14%. After paying the taxes, you spend 10% of your net income to buy new clothes and other accessories for the next school year and 1% to buy school supplies. After buying clothes and school supplies, you use 25% of the remaining money to buy savings bonds. For each dollar you spend to buy savings bonds, your parents spend $0.50 to buy additional savings bonds for you. Write a program that prompts the user to enter the pay rate for an hour and the number of hours you worked each week. The program then outputs the following:

a. ***Your income before and after taxes from your summer job***
b. ***The money you spend on clothes and other accessories***
c. ***The money you spend on school supplies***
d. ***The money you spend to buy savings bonds***
e. ***The money your parents spend to buy additional savings bonds for you***

Lab Task 7

A cricket game is to be held in a stadium and there are four seating categories available for the audience. Class A seats cost $20, Class B seats cost $15, Class C seats cost $10, and Class D seats cost $5. You should write a JAVA program that asks how many tickets for each class of seats were sold and finally display the total income generated and income corresponding to ticket sales.

Lab Task 8

Write a program that reads an integer between 0 and 1000 and adds all the digits in the integer. For example, if an integer is 932, the sum of all its digits is 14.
Enter a number between 0 and 1000: 999
The sum of the digits is 27

Lab Task 9

Consider the statements:

```
double x = 75.3987;
double y = 982.89764;
```

What is the output of the following statements?

```
System.out.printf("%.2f %n", x);
System.out.printf("%.2f %n", y);
System.out.printf("%.3f %n", x);
System.out.printf("%.3f %n", y);
```

Lab Task 10

Write JAVA statements using **`System.out.prinf()`** *statement to display output as given below*

```
Degrees   Radians   Sine      Cosine    Tangent
30        0.5236    0.5000    0.8660    0.5773
60        1.0472    0.8660    0.5000    1.7320
```

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

      // Display the header of the table using System.out.printf()

      int degrees = 30;
      double radians = Math.toRadians(degrees);
      double sin = Math.sin(radians);
      double cos = Math.cos(radians);
      double tan = Math.tan(radians);

      // Display the Data of the table using System.out.printf()

      degrees = 60;
      radians = Math.toRadians(degrees);
      sin = Math.sin(radians);
      cos = Math.cos(radians);
      tan = Math.tan(radians);

      // Display the Data of the table using System.out.printf()
   }
}
```