

## Unit 1: Primitive Types

# Topic 4: Working with User Input

Name: \_\_\_\_\_

### Programming Challenges

*Work with your partner to solve these!*

The demo code for today is in the slides.

1. Write a program that welcomes the user by name (so you will need to ask for their name first!) and then asks them to input 3 different numbers, one at a time. Your program should then print the average of those numbers. The user should be able to provide either an integer or a decimal value for *any* (or all) of the three values.

Here is sample input/output for the program (input underlined in **yellow**)

```
CONSOLE  SHELL
Please enter your name: Mr. Miller
Welcome Mr. Miller!
Please enter the first number: 12
Please enter the second number: 15.6
Please enter the third number: 25
The average of 12.0, 15.6, and 25.0 is 17.533333333333335
```

Here is another example (running the program a second time):

```
Please enter your name: Mr. Miller
Welcome Mr. Miller!
Please enter the first number: 7.1
Please enter the second number: 12
Please enter the third number: 19.45
The average of 7.1, 12.0, and 19.45 is 12.85
```

[Hints!](#)

**Copy your program's code from and paste it below:**

(use the **Courier New** font for code-style)

**Partner Testing!** *Trade and test each other's*

Did your partner find any bugs in your program?

programs! Make sure your partner's program meets the requirements, and your partner will do the same for you!

Give up? (**Don't give up!**) or if you wish to compare: [Sample solution](#)

**Before** trying problem 2, delete all code from your main method, and **copy/paste** this code then run it and study the results:

```
System.out.println("1" + "2" + "3");           // line 1
System.out.println(1 + 2 + 3);                 // line 2
System.out.println("Now look at this: " + 1 + 2 + 3); // line 3
System.out.println("Now this: " + (1 + 2 + 3)); // line 4
```

**a.** Why do you suppose the sum in **line 3** looks *more like* the sum on line 1 than the sums on lines 2 or 4?

a.

**b.** Why does adding ( ) around the sum in line 4 cause it to behave differently than line 3?

b.

[Check answer](#)

**Delete the code from your main method, *or* comment it all out if you want to keep it.**

- 2.** Now, write a program that asks the user for any 3-digit integer, then prints that number in reverse. For example, the user should be able to provide 759 and the program should print 957. If the user provides a number like 270, it prints 072 (not 72).

Here are **two** sample input/output after running the program **twice**:

CONSOLE	SHELL	CONSOLE	SHELL
Please enter a 3-digit integer: 759 The reverse of 759 is: 957 →		Please enter a 3-digit integer: 270 The reverse of 270 is: 072 →	

[Hints!](#)

**Copy your program's code from and paste it below:**

(use the **Courier New** font for code-style)

**Partner Testing!** Trade and test each other's programs! Make sure your partner's program

Did your partner find any bugs in your program?

meets the requirements, and your partner will do the same for you!

Give up? (**Don't give up!**) or if you wish to compare: [Sample solution](#)

**Delete the code from your main method again, or comment out.**

3. Write a program that asks for Player 1's name, followed by an integer, then asks for Player 2's name, followed by a *second* integer (different from the first). Print out an appropriate message depending on which Player's number is larger, as shown in the examples below:

Player 1, enter your name: <u>Adam</u>	Player 1, enter your name: <u>Adam</u>
Enter your number: <u>34</u>	Enter your number: <u>12</u>
Player 2, enter your name: <u>Janice</u>	Player 2, enter your name: <u>Janice</u>
Enter your number: <u>56</u>	Enter your number: <u>8</u>
Janice's number is bigger!	Adam's number is bigger!

**Note!** This requires the use of an **if-else** statement with a **condition**, which we haven't yet talked about in Java -- we will talk about this in depth later, but for now, here is the general structure of an if-else statement in Java:

```
if (condition) {  
    // do something  
} else {  
    // do something else  
}
```

You figure out the rest

**Copy your program's code from and paste it below:**

(use the **Courier New** font for code-style)

Give up? (**Don't give up!**) or if you wish to compare: [Sample solution](#)

4. **Challenging:** Write a program that asks the user for any 5-digit integer not starting with 0, increments all digits of that number by 1, then prints the new number. For

digits that are 9, the program should increment to 0. For example, the user should be able to provide 25497 and the program should print 36508.

*Here is sample input/output for the program:*

```
CONSOLE SHELL
Type in a 5-digit number: 25497
The digits of 25497 incremented by 1 are 36508
→
```

[Hints!](#)

**Copy your program's code from and paste it below:**

(use the **Courier New** font for code-style)

***Sample solution for the above challenge will be posted tomorrow!***

**Done!**

Submit in Google Classroom:

Turn in

# HINTS

Hints for Problem 2 ([back](#)):

- Use `nextInt()` to obtain an `int` from the user.
- You can use the `%` and `/` operators creatively to parse out the three digits, and store each digit in its own variable (since you are working with ints, recall that `/` truncates!)

Examples:

```
654 / 100 = 6
654 % 100 = 54
(654 % 100) / 10 = 5
```

Can you figure out how to get the “4” ?

- What you learned in the little exploration before problem 2 involving this line of code might be helpful...

```
System.out.println("Now look at this: " + 1 + 2 + 3);
```

Hints for Problem 3 ([back](#)):

- You could create and use five different variables, one to store each digit as you figure it out (digit1, digit2, etc.)

Here are some math tips you might find useful!

- $38654 / 10000 = 3$
- $38654 \% 10 = 4$
- $(8 + 1) \% 10 = 9$                        $// 9 \% 10 = 9$  since 10 doesn't fit into 10, so the remainder is 9!
- $(9 + 1) \% 10 = 0$

## Hints for Problem 1 ([back](#))

Since the user should be able to enter an integer *or* a decimal for *any* of the three values, use the `nextDouble()` method -- rather than `nextInt()` -- three different times to obtain all three numbers from the user (because you can store an `int` value in a `double` variable, but you *can't* store a `double` value in an `int` variable!)

## Sample Solution Problem 1 ([back](#))

Here is one way to do it -- yours may look different, and that's OK!

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

System.out.print("Please enter your name: ");
String name = scan.nextLine();
System.out.println("Welcome " + name + "!");

System.out.print("Please enter the first number: ");
double num1 = scan.nextDouble();
scan.nextLine();

System.out.print("Please enter the second number: ");
double num2 = scan.nextDouble();
scan.nextLine();

System.out.print("Please enter the third number: ");
double num3 = scan.nextDouble();
scan.nextLine();

double average = (num1 + num2 + num3) / 3;
System.out.println("The average of " + num1 + ", " + num2 + ", and " + num3 + " is " + average);
```



Answer ([back](#))

```
System.out.println("1" + "2" + "3");           // line 1
System.out.println(1 + 2 + 3);                 // line 2
System.out.println("Now look at this: " + 1 + 2 + 3); // line 3
System.out.println("Now this: " + (1 + 2 + 3)); // line 4
```

Printed output:

```
123
6
Now look at this: 123
Now this: 6
```

**a.** Why do you suppose the sum in **line 3** looks *more like* the sum on line 1 than the sums on lines 2 or 4?

**b.** Why does adding ( ) around the sum in line 4 cause it to behave differently than line 3?

**a.** The initial part of the println method's argument is a string ("Now look at this: ") followed by a + and a 1 so Java converts 1 to a String "1" and *concatenates* it to the first part; it does the same thing for the 2 and 3, converting the to a String "2" and a String "3" and concatenating them, resulting in **"Now look at this: 123"**

**b.** When Java looks at the argument of the println method, it sees there a group in parentheses: (1 + 2 + 3), so it evaluates that *first* and since those are all ints, it sums them up to 6. *Then* it takes the initial part of the argument, "Now this: " and concatenates it with "6" (converted from int to a String)

## Sample Solution Problem 2 ([back](#))

Here is one way to do it -- yours may look different, and that's OK!

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

System.out.print("Please enter a 3-digit integer: ");
int userNum = scan.nextInt();
scan.nextLine();

// parse out the three different digits
int hundreds = userNum / 100; // int division truncates
int tens = (userNum % 100) / 10;
int ones = (userNum % 100) % 10;

System.out.println("The reverse of " + userNum + " is: " + ones + tens + hundreds);
```

Note that this print statement:

```
System.out.println("The reverse of " + userNum + " is: " + ones + tens + hundreds);
```

Does *not* actually add the three numbers together before printing them out, but rather prints them separately as 3 strings (this is because they are being appended to a string; see the short exploration before problem 2 for more info!) -- using this strategy, you can easily print a number like 072 (rather than 72).

## Sample Solution Problem 3 ([back](#))

Here is one way to do it -- yours may look different, and that's OK!

```
1  import java.util.Scanner;
2
3  public class FiveDigits
4  {
5      public static void main(String args[])
6      {
7          Scanner scan = new Scanner(System.in);
8
9          System.out.print("Type in a 5-digit number: ");
10         int userNum = scan.nextInt();
11
12         int d1 = (userNum % 10 / 1 + 1) % 10;
13         int d2 = (userNum % 100 / 10 + 1) % 10;
14         int d3 = (userNum % 1000 / 100 + 1) % 10;
15         int d4 = (userNum % 10000 / 1000 + 1) % 10;
16         int d5 = (userNum % 100000 / 10000 + 1) % 10;
17
18         System.out.println("The digits of " + userNum + " incremented by 1 are " + d5 + d4 + d3 + d2 + d1);
19     }
20 }
```

## Sample Solution ([back](#))

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

System.out.print("Player 1, enter your name: ");
String player1 = scan.nextLine();

System.out.print("Enter your number: ");
int num1 = scan.nextInt();
scan.nextLine();

System.out.print("Player 2, enter your name: ");
String player2 = scan.nextLine();

System.out.print("Enter your number: ");
int num2 = scan.nextInt();
scan.nextLine();

if (num1 > num2) {
    System.out.println(player1 + "'s number is bigger!");
} else {
    System.out.println(player2 + "'s number is bigger!");
}
```