

Chapter 2 Worksheet

1. **What is a variable?**

A variable is a named storage location in a computer's memory that holds a value. It is used to store and manipulate data in a program.

2. **Why use a variable?**

Variables are used to store and manage data in a program. They allow us to assign values, perform calculations, and manipulate data in a flexible and dynamic manner. Using variables also helps improve code readability and maintainability.

3. **Can a variable be reassigned?**

Yes, a variable can be reassigned with a new value during the execution of a program. This means you can change the value stored in a variable multiple times throughout the code.

4. **What is the structure of declaring a variable?**

The structure of declaring a variable typically involves specifying the data type followed by the variable name. For example:

`<datatype> <variable_name>;` Here, `<datatype>` represents the type of data the variable will hold, and `<variable_name>` is the chosen identifier for the variable.

5. **What is the structure for assigning a variable?**

The structure for assigning a variable involves using the assignment operator (`=`) to assign a value to the variable. For example:

`<variable_name> = <value>;` Here, `<variable_name>` is the name of the variable, and `<value>` is the data you want to assign to the variable.

6. **Which data types can hold whole values?**

Data types such as `int` (integer), `long` (long integer), `short` (short integer), and `byte` can hold whole values. These data types represent integers without any decimal places.

7. **Which data types can hold decimal values?**

Data types such as `float` (single-precision floating-point), `double` (double-precision floating-point), and `decimal` (decimal floating-point) can hold decimal values. These data types can represent numbers with decimal places.

8. **Can a double data type hold a whole number?**

Yes, a double data type can hold a whole number. It can store both whole numbers and numbers with decimal places. For example, 5 and 5.0 can be stored in a double variable.

9. **Can an int data type hold a number that has a decimal value?**

No, an `int` data type cannot hold a number with a decimal value. The `int` data type is designed to store whole numbers only. If you attempt to assign a number with a decimal value to an `int` variable, the decimal part will be truncated.

10. **Declare a variable to hold a whole number with the identifier of num.**

```
int num;
```

11. **Assign num with the value of 102.**

```
num = 102;
```

12. **Change the value of num to 202.**

```
num = 202;
```

13. **Declare and assign a variable with the identifier decNum with the value of 102.777.**

```
double decNum = 102.777;
```

14. **What variable is needed to be able to read in values?**

To read in values from the user or an input source, you typically use the `Scanner` variable.

15. **Does that variable have to be called (have the identifier of) input?**

No, the `Scanner` variable doesn't have to be called "input." You can choose any valid identifier as long as it follows the naming conventions.

16. **What library has to be imported to use the Scanner variable?**

To use the `Scanner` class, you need to import the `java.util` library. The import statement at the top of the code file would look like this:

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

17. What methods in the Scanner class read in whole numbers?

The Scanner class provides methods such as `nextInt()` and `nextLong()` to read in whole numbers. These methods can be used to read and parse integers and long integers respectively.

18. What methods in the Scanner class read in decimal numbers?

The Scanner class provides methods such as `nextFloat()` and `nextDouble()` to read in decimal numbers. These methods can be used to read and parse floating-point numbers with single or double precision.

19. How do the identifiers have to be for constants?

Identifiers for constants are usually written in uppercase letters with underscores (`_`) between words. For example: `final int MAX_VALUE = 100;` Here, `MAX_VALUE` is a constant identifier.

20. What are the two ways of doing naming conventions for identifiers that have more than one word?

Two common naming conventions for identifiers with multiple words are:

CamelCase: In this convention, the first letter of each word, except the first word, is capitalized, and there are no spaces or underscores between words. `myVariable`, `calculateSum`, `numberOfStudents`.

Underscore-separated: In this convention, words are separated by underscores (`_`), and all letters are lowercase. For example: `my_variable`, `calculate_sum`, `number_of_students`.

Both conventions are widely used. The choice between them often depends on personal preference.

21. What kind of data type would be appropriate for the following identifiers:

<code>VALUE_OF_DIMES</code>	<code>double</code>
<code>howManyDimes</code>	<code>Int</code>
<code>typesOfBoxes</code>	<code>Array</code>
<code>TheProgram</code>	a class
<code>semester</code>	<code>String</code>
<code>balance</code>	<code>double</code>
<code>color</code>	<code>String</code>
<code>pounds</code>	<code>double</code>

22. If given `int x = 2` and `int y = 5`, what is the output from the following Java statements?

`System.out.println(y/x);` Output: 2

`double z = x/y;`

`System.out.println(z);` Output: 0.4

`int q = x/y;`

`System.out.println(q);` Output: 2

23. If given `int x = 2` and `double y = 5`, what is the output from the following Java statements?

`System.out.println(y/x);` Output: 2.5

`double z = x/y;`

`System.out.println(z);` Output: 0.4

`int q = x/y;`

`System.out.println(q);` Output: 0

24. If given `double x = 2` and `double y = 5`, what is the output from the following Java statement?

`System.out.println(y/x);` Output: 2.5

25. Write the Java statement for 2^8 .

`double result = Math.pow(2, 8);`

26. What is the output of the following Java statements?

a. `System.out.println(5%2);` Output: 1

b. `System.out.println(2%5);` Output: 2

27. What is the output of the following Java statements?

a. `int i = 2; i += 5; System.out.println(i);` Output: 7

b. `int i = 4; i -= 2; System.out.println(i);` Output: 2

c. `int i = 4; i *= 2; System.out.println(i);` Output: 8

d. `int i = 10; i /= 2; System.out.println(i);` Output: 5

e. `int i = 16; i %= 2; System.out.println(i);` Output: 0

f. `int i = 2; i -= 2; System.out.println(i);` Output: 0

28. What is the output of the following Java statements?

a. `int i = 102; System.out.println(++i);` Output: 103

b. `int i = 102; System.out.println(--i);` Output: 101

c. `int i = 102; System.out.println(i++);` Output: 102

d. `int i = 102; System.out.println(i--);` Output: 102

e. `int i = 102; int num = 10 * i++;`
`System.out.println("i is " + i + " and num is " + num);` Output: i is 103
and num is 1020

f. `int i = 102; int num = 10 * ++i;`
`System.out.println("i is " + i + " and num is " + num);` Output: i is 103
and num is 1030

29. What is the output of the following Java statements?

a. `System.out.println((int) 3.6);` Output: 3

b. `System.out.println((double) 6/3);` Output: 2.0

c. `System.out.println(7/2.0);` Output: 3.5

d. `int num = 0; num += 4.5; System.out.println(num);` Output: 4. (The fractional part is truncated since `num` is an `int`)

e. `double num1 = 8.2; int num2 = num1;`
`System.out.println("num1 is " + num1 + " and num2 is " + num2);` Output:
num1 is 8.2 and num2 is 8 (The fractional part is truncated during the assignment to `num2`)

30. Write the Java statement that will display the variable `course`, which holds the value of 102.654, as 102 – without changing the value of, declaring a new variable, or just printing the value of 102.

`double course = 102.654; System.out.println((int) course);` // Output: 102

31. Giving the `int` variable `upper`, and the `int` variable `lower`, write the Java statement that will display `upper / lower` with the whole, correct value.

`int upper = 10; int lower = 3; System.out.println(upper / lower);` // Output: 3 (integer division, resulting in the whole number quotient)