Java Assignment 2

Jamal Azad

05/21/22

1. What do you understand by variable?
   In programming, a variable is **a value that can change, depending on conditions or on information passed to the program**. Typically, a program consists of instructions that tell the computer what to do and data that the program uses when it is running.

2. Name types of variables in java.

   In Java there are four types of variables:

   a) Non-static fields or Instance Variables
   b) Static fields.
   c) Local variables.
   d) Parameters.

   ➔ A non-static field or instance variable is a variable that belongs to an object. Objects keep their internal state in non-static fields. Non-static fields are called instance variables, because they belong to instances (objects) of a class. Non-static fields are covered in more detail in the text on Java fields.

   ➔ A static field is a variable that belongs to a class. A static field has the same value for all objects that access it. Static fields are also called class variables. Static fields are also covered in more detail in the text on Java fields.

   ➔ A local variable is a variable declared inside a method. A local variable is only accessible inside the method that declared it. Local variables are covered in more detail in the text on Java methods.

   ➔ A parameter is a variable that is passed to a method when the method is called. Parameters are also only accessible inside the method that declares them, although a value is assigned to them when the method is called. Parameters are also covered in more detail in the text on Java methods.

3. What is the difference between static and instance variables?
   Instance variables are created when an object is created with the use of the keyword 'new' and destroyed when the object is destroyed. Static variables are created when the program starts and destroyed when the program stops. Instance variables can be accessed directly by calling the variable name inside the class.

4. Which type of variable requires a keyword to determine its scope?
   Static variable requires a keyword to determine its scope.
   When we define a variable with a static keyword inside the class, its scope is within the class. That is, the scope of a static variable is within the class. All the methods, constructors, and blocks inside the class can access static variables by using the class name.

5. Which thing specifies the size and type of variable?
   A variable declaration specifies the size and type of variable.
6. Can we use a variable without declaring it?
   Variables are containers for storing data values. So, Answer is NO. A variable is only a name given to a memory location, all the operations done on the variable effects that memory location. In Java, **all the variables must be declared before using**.

7. How can we declare a variable?
   To create a variable, you must specify the type and assign it a value:
   Syntax
   ```
   type variableName = value;
   ```

8. How can we initialize a variable?
   The syntax for an initializer is the type, followed by the variable name, followed by an equal sign, followed by an expression. That expression can be anything, provided it has the same type as the variable. In this case, the expression is 10, which is an int.

9. Make a list of common Java data types.
   Data types are divided into two groups:
   a) Primitive Data
   b) Non-primitive Data

---

a) There are eight primitive data types in Java:
byte, short, int, long, float, double, boolean and char

Data Type	Size	Description
byte	1 byte	Stores whole numbers from -128 to 127
short	2 bytes	Stores whole numbers from -32,768 to 32,767
int	4 bytes	Stores whole numbers from -2,147,483,648 to 2,147,483,647
long	8 bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4 bytes	Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits
double	8 bytes	Stores fractional numbers. Sufficient for storing 15 decimal digits
boolean	1 bit	Stores true or false values
char	2 bytes	Stores a single character/letter or ASCII values

b) Non-primitive data types - such as String, Arrays and Classes.
Non-primitive data types are called reference types because they refer to objects.

10. Which data types are created by programmers and are not defined by Java?
    Non-primitive data are created by programmers.
    Class, object, array, string, and interface are called non-primitive data types in Java. These data types are not predefined in Java. So, these are created by programmers.

11. Java has how many primitive data types?
    There are eight primitive data types in Java:
    1. byte, 2. short, 3. int, 4. long, 5. float, 6. double, 7. boolean and 8. char
12. What data type would you use for storing the number of students in a class?
    long int , because students number can increase or decrease and *long int* can accommodate these changes.

13. What data type would you use for storing the average test score in a class?
    double, because average test score can be fractional number and *double* can accommodate these changes.

14. How would you declare a variable storing the tax rate?
    `double taxRate = 7.25;`

15. How would you declare a variable that tells the grade (A, B, C, or D) of students?
    To declares variable with single character, need to use char.
    Syntex: `char StudentsGrade = 'A';`

16. List the name conventions you learned in class about variables.
    Class names should be nouns, in mixed case with the first letter of each internal word capitalized. Try to keep your class names simple and descriptive.

17. Name the Data type used for an object 'Pen'.
    It has to be *char*.

18. Write a program and declare the variable of each data type you learned in class. Initialize each variable with appropriate values and print them.

```
public class Variable{

   public static void main(String[] args) {
        System.out.println( "This is about variable of data types in
Java!" );
        System.out.println(" 1) String - stores text, such as
'Hello'");
        System.out.println(" 2) int - stores integers (whole
numbers), without decimals, such as 123 or -123. ");
        int value1= 123;
        System.out.println(value1);
        System.out.println(" 3) Double - stores floating point
numbers, with decimals, such as 19.99 or -19.99." );
        double value2= 19.99;
        System.out.println(value2);
        System.out.println(" 4) char - stores single characters, such
as 'a' or 'B'.");
        char value3= 'A';
        System.out.println(value3);
        System.out.println("long : 8 bytes Stores whole numbers from
-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 ");
        long value4= 922337203;
        System.out.println(value4);
        System.out.println("5) boolean - stores values with two
states: true or false. ");
        boolean value5= true ;
        System.out.println(value5);
        boolean value6= false ;
        System.out.println(value6);
```

19. Write a program to print your name, email, address, phone number, and id.

```
public class MyInfo {
public static void main(String[] args) {
      System.out.println("My Information");
      System.out.println("Name: Jamal Azad");
      System.out.println("My Email: Jamal@email.com");
      System.out.println("My Phone: 646-123-4567");
      System.out.print("My ID Number: ");
      int myId = 001;
      System.out.println(myId);
     }
}
```

20. What do you understand by typecasting?

Type casting is when you assign a value of one primitive data type to another type.
In Java, there are two types of casting:
a) **Widening Casting** (automatically) - converting a smaller type to a larger type size
byte -> short -> char -> int -> long -> float -> double
b) **Narrowing Casting** (manually) - converting a larger type to a smaller size type
double -> float -> long -> int -> char -> short -> byte

21. In which type casting Java automatically converts one data type to another data type?

Widening Type Casting or Implicit Type Conversion or Automatic type conversion is a process of automatic conversion of one data type to another by the compiler, without involving the programmer.
This process is called Widening Conversion because the compiler converts the value of narrower (smaller size) *data type* into a value of a broader (larger size) data type without loss of information. The implicit data type conversion is possible only when:

- The two data types are compatible with each other.
- There is a need to convert a smaller or narrower data type to the larger type size.

22. Write a program to convert int to double and print a message stating which type casting you did?

```
public class WideningCast{

public static void main(String[] args) {

            int valueInt = 11;
            double valueDouble = valueInt;
            System.out.print("Integer value was " );
            System.out.println(valueInt);
            System.out.print("Value after converting into double
using Widening Casting: ");
            System.out.println(valueDouble);

        }
}
```