Islamic University of Technology

Visual Programming Lab

CSE 4402

Introduction to Java

Author:
Ishmam Tashdeed
Sabrina Islam
CSE, IUT

Contents

1 Variables

Variables are containers for storing data values. These are similar to C/C++. In Java, there are different types of variables, but the most used ones are:

- **String** - stores text, such as “Hello”. String values are surrounded by double quotes

- **int** - stores integers (whole numbers), without decimals, such as 123 or -123

- **float** - stores floating point numbers, with decimals, such as 19.99 or -19.99

- **double** - stores floating point numbers with more precision, with decimals, such as 19.99628249 or -19.99725265

- **char** - stores single characters, such as ‘a’ or ‘B’. Char values are surrounded by single quotes

- **boolean** - stores values with two states: `true` or `false`

They can be initialized and used as:

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

    int x = 123;
    double y = 3.14;
    boolean z = true;
    char symbol = ’@’;
    String name = "Bro";
  }
}
```

2 Comments in Java

For single-line comments, we use `//` and for multi-line comments, we use `/* */`

3 Taking Input

For taking input from the user we have to use the `Scanner` class.

```
import java.util.Scanner;

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

    Scanner scanner = new Scanner(System.in);

    System.out.println("What is your name? ");
    String name = scanner.nextLine(); // Taking String as
   input

    System.out.println("How old are you? ");
    int age = scanner.nextInt(); // Taking int as input
    scanner.nextLine();

    System.out.println("What is your favorite food?");
    String food = scanner.nextLine();

    System.out.println("Hello " + name);
    System.out.println("You are " + age + " years old");
    System.out.println("You like " + food);

    scanner.close(); // Closing the scanner
  }
}
```

It is best practice to always close the scanner using `scanner.close()`.

4 Operators

Operators are used to perform operations on variables and values. Java divides the operators into the following groups:

- Arithmetic operators (+, -, *, /, %)
- Assignment operators (=, +=)
- Comparison operators (==, <, >, >=, <=, !=)
- Logical operators (&&, ||)

- Bitwise operators (&, |, >>, <<)

These are the same as seen in C/C++.

5 Strings

A String variable contains a collection of characters surrounded by double quotes. Create a string variable using:

```
String greeting = "Hello";
```

Using the + sign, multiple strings can be concatenated.

```
String firstName = "John";
String lastName = "Doe";
System.out.println(firstName + " " + lastName);
```

There are multiple built-in functions for String class such as:

- Find string length using the length() method.
- To convert to uppercase or lowercase use toUpperCase() and toLowerCase().
- To find a specific character or substring in a string use indexOf().

In Java, string objects are **immutable**. Immutable simply means unmodifiable or unchangeable. Once a string object is created its data or state can't be changed but a new string object is created.

```
String first_name = "Ishmam";
full_name = first_name.concat(" Tashdeed");
System.out.println(first_name); // Only first name
System.out.println(full_name); // Full name

```

6 Conditionals

The conditional block syntax in Java is the same as C/C++.

```
int time = 22;
if (time < 10) {
  System.out.println("Good morning.");
} else if (time < 18) {
  System.out.println("Good day.");
} else {
  System.out.println("Good evening.");
}
```

7 Loops

Loops work similar to how they behave in C/C++. Some examples of loops are:

- `for` loop:

  ```
  for(int i = 0; i < 10; i++){
      System.out.println(i);
  }
  ```

- `while` loop:

  ```
  int i = 0;
  while(i < 10){
      System.out.println(i);
      i++;
  }
  ```

- `do while` loop:

  ```
  int i = 0;
  do{
      System.out.println(i);
      i++;
  } while(i < 10);
  ```

8 Arrays

Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.

```
// initializing array
int[] arr = { 1, 2, 3, 4, 5 };

// size of array
int n = arr.length;
```

```
// traversing array
for (int i = 0; i < n; i++){
    System.out.print(arr[i] + " ");
}
```

9 Object Oriented Programming

Object-Oriented Programming (OOP) is a methodology or paradigm to design a program using classes and objects. It simplifies software development and maintenance by providing some concepts. The main pillars of OOP are:

- Inheritance
- Polymorphism
- Abstraction
- Encapsulation

9.1 Class

In object-oriented programming, a class is a blueprint from which individual objects are created. Using classes, you can create multiple objects with the same behavior instead of writing their code multiple times. This includes classes for objects occurring more than once in your code. An example of creating a class in Java is:

```
// Create a Student class

public class Student {
   // Declaring attributes
   String name;
   int rollNo;
   String section;

   // initialize attributes in constructor
   Student(String name, int rollNo, String section){
      this.name= name;
      this.rollNo = rollNo;
      this.section = section;
   }
   //Getter methods
   public String getName() {
       return this.name;
   }

   public int getRoll() {
       return this.rollNo;
   }

   public String getSection() {
       return this.section;
   }

   // print details
   public void printDetails() {
      System.out.println("Student Details:");
      System.out.println(this.getName());
      System.out.println(this.getRoll());
      System.out.println(this.getSection());
   }
}
```

In Java, if you do not specify an access modifier, the member (class, method, or variable) gets **private** access by default.

9.2 Object

An Object is a basic unit of Object-Oriented Programming that represents real-life entities. A typical Java program creates many objects, which as you know, interact by invoking methods. The objects are what perform your code, they are the part of your code visible to the viewer/user. An example of creating an object in Java is:

```
// Create student objects
Student student1 = new Student("Robert", 1, "A");

// Print student details
student1.printDetails();
```

10 Tasks

10.1 Task 1

Design a mini console quiz game where the player is asked 5 math questions. Track how many correct answers they give and display a score at the end.

10.2 Task 2

Build a Movie Rental System where you can,

- Add movies. Each movie will have title, name of the director, name of the producer, list of actors/actresses, genre, and a unique ID.
- Search for a movie by title or genre.
- Users can rent a movie (only if it is available). Each user must have an ID and a name.
- View all available movies.
- View movies currently rented by a user.
- Prevent the same user from renting the same movie twice.