Introduction to Java for C++ Programmers

Segment - 2

JAC 444

By: Mahboob Ali

Objectives

Upon completion of this segment you should be able to:

- Write your first java program.
- Class: The structure of java program.
- Basics about:
  - Build-in primitives
  - Commentss
  - Flow of control

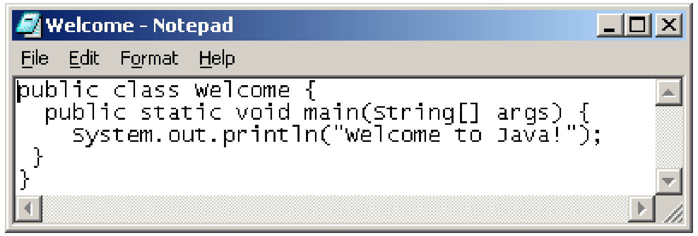
Simple Java Program

```
//This program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Method Block

Class Block

Creating, Compiling, and Running Programs



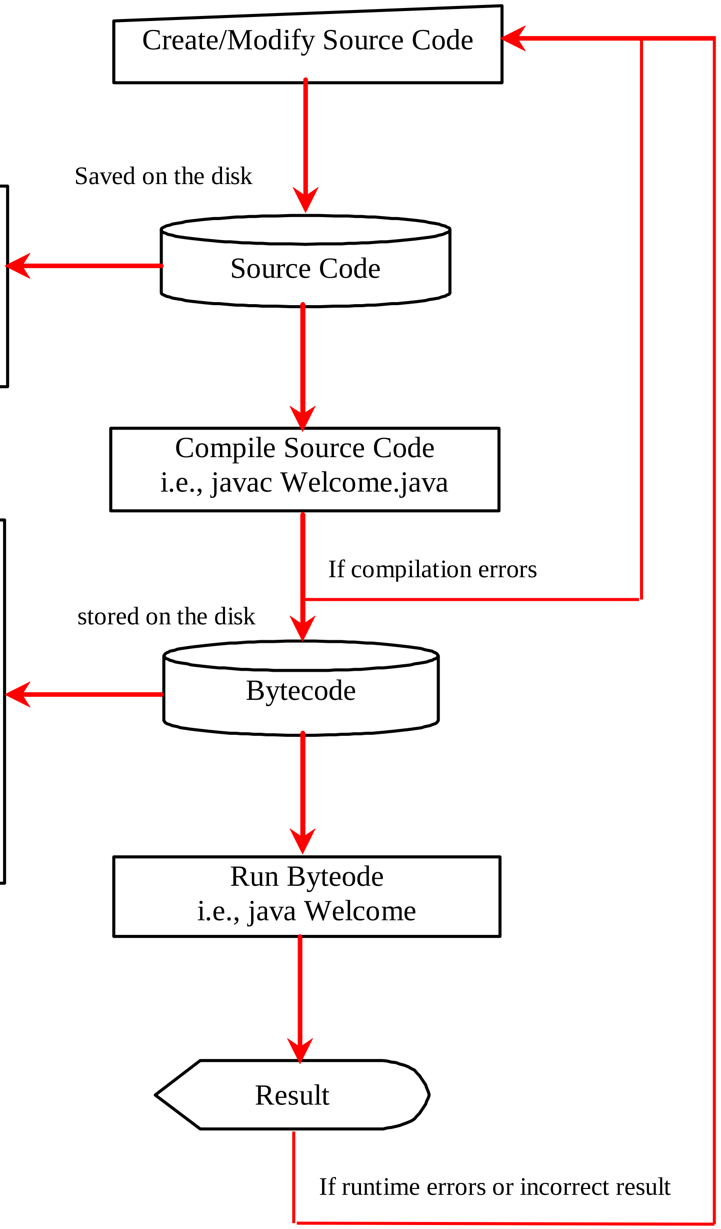
```
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Source code (developed by the programmer)

```
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Byte code (generated by the compiler for JVM to read and interpret, not for you to understand)

```
…
Method Welcome()
  0 aload_0
  …

Method void main(java.lang.String[])
  0 getstatic #2 …
  3 ldc #3 <String "Welcome to
Java!">
  5 invokevirtual #4 …
  8 return
```



Create/Modify Source Code

Saved on the disk

Source Code

Compile Source Code
i.e., javac Welcome.java

If compilation errors

stored on the disk

Bytecode

Run Byteode
i.e., java Welcome

Result

If runtime errors or incorrect result

Programming Errors

- Syntax Errors:
- Runtime Errors.
- Logic Errors.

Syntax Errors

- Errors that are detected by the complier are called syntax or compile errors.
- Syntax errors result from errors in code construction.
  - Mistyping a keyword.
  - Omitting some necessary punctuation.
  - Using an opening brace without a corresponding closing brace.

Runtime Errors

- Runtime errors are errors that cause a program to terminate abnormally.
- Input mistakes typically causes runtime errors.
  - User entered a string value instead of number input.
  - Division by zero.

Logic Errors

- Logical errors occur when a program does not perform the way it was intended to.
  - ((9 / 5) * 35 + 32); suppose to convert Celsius 35 degree to Fahrenheit. You will get the answer 67 Fahrenheit, which is wrong. It should be 95 Fahrenheit. Because in Java division of integer results in a integer value, factorial part is truncated

Trace a Program Execution

Enter main method

```
//This program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

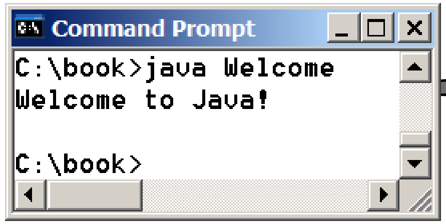
Trace a Program Execution

Execute statement

```
//This program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Trace a Program Execution

```
//This program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```



```
C:\book>java Welcome
Welcome to Java!

C:\book>
```

print a message to the console

Classes

- The **classes** are the essential Java constructs.
- Every Java program must have at least one class.
- By convention, class names start with an uppercase letter.

```java
// This program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Methods

- What is System.out.println?
- It is a method: a <u>*collection of statements*</u> that performs a sequence of operations to display a message on the console.
- It can be used even without fully understanding the details of how it works.
- Use a string argument.

Computing the Area of a Circle

```
public class ComputeArea{
    public static void main (String[] args) {
            double radius;
            double area;
            // Step 1 : Read in Radius
            // Step 2 : Compute Area
            // Step 3 : Display the area
    }
   }
```

- The next step is to define some values for the radius,

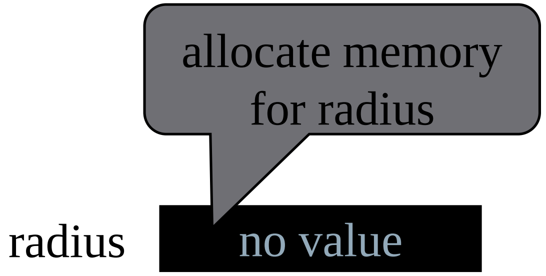
```
public class ComputeArea{
public static void main (String[] args) {
        double radius;            //Declare radius
        double area;              //Declare area
```

```
            // Assign  a radius

            radius = 20; // radius is now 20

            / / Compute area

            area = radius * radius * 3.14159;


            // Display results

System.out.println(“The are for the circle of radius ” +
radius + “ is ” + area);

  }

}
```

- Every variable has a name, size, type and value.
- The plus sign (+) has two meanings: one for addition and the other for concatenating (combining) stings. String concatenation operator, combines two strings into one.

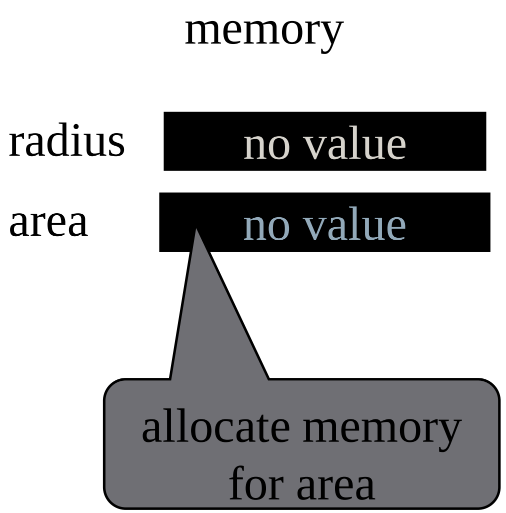
Trace a Program Execution

```
public class ComputeArea {
  /** Main method */
  public static void main(String[] args) {
    double radius;
    double area;

    // Assign a radius
    radius = 20;

    // Compute area
    area = radius * radius * 3.14159;

    // Display results
    System.out.println("The area for the circle of radius " +
      radius + " is " + area);
  }
}
```



Trace a Program Execution

```
public class ComputeArea {
  /** Main method */
  public static void main(String[] args) {
    double radius;
    double area;

    // Assign a radius
    radius = 20;

    // Compute area
    area = radius * radius * 3.14159;

    // Display results
    System.out.println("The area for the circle of radius " +
      radius + " is " + area);
  }
}
```



memory

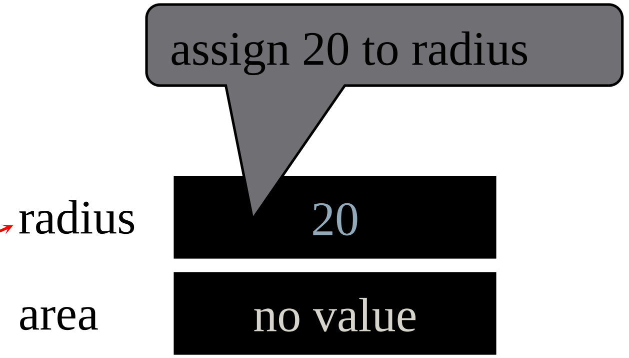
radius no value

area no value

allocate memory for area

Trace a Program Execution

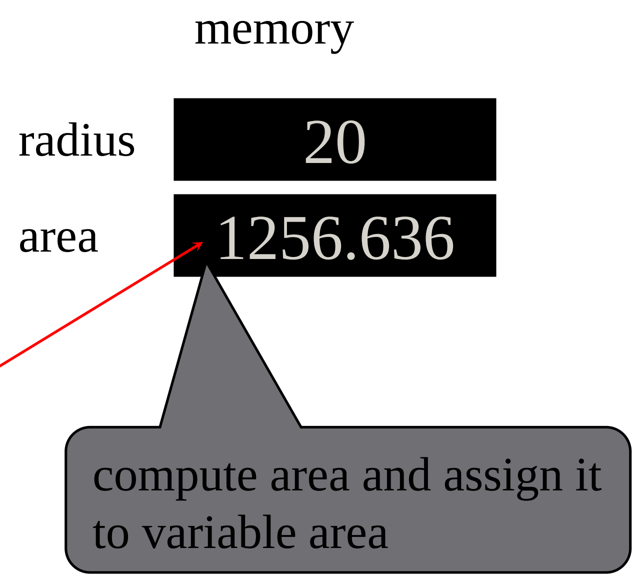
```
public class ComputeArea {
  /** Main method */
  public static void main(String[] args) {
    double radius;
    double area;

    // Assign a radius
    radius = 20;

    // Compute area
    area = radius * radius * 3.14159;

    // Display results
    System.out.println("The area for the circle of radius " +
      radius + " is " + area);
  }
}
```



assign 20 to radius

Trace a Program Execution

```
public class ComputeArea {
  /** Main method */
  public static void main(String[] args) {
    double radius;
    double area;

    // Assign a radius
    radius = 20;

    // Compute area
    area = radius * radius * 3.14159;

    // Display results
    System.out.println("The area for the circle of radius " +
      radius + " is " + area);
  }
}
```



compute area and assign it to variable area

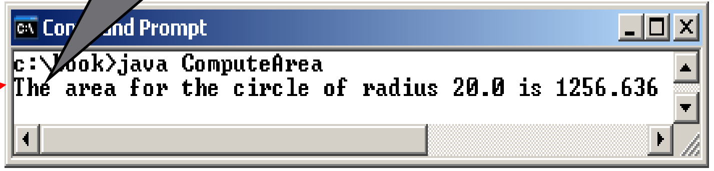
Trace a Program Execution

```
public class ComputeArea {
  /** Main method */
  public static void main(String[] args) {
    double radius;
    double area;

    // Assign a radius
    radius = 20;

    // Compute area
    area = radius * radius * 3.14159;

    // Display results
    System.out.println("The area for the circle of radius " +
      radius + " is " + area);
  }
}
```

memory

radius	20
area	1256.636

print a message to the console



```
c:\book>java ComputeArea
The area for the circle of radius 20.0 is 1256.636
```

Reading Input from the Console

1. Create a Scanner object

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

2. Use the methods next(), nextByte(), nextShort(), nextInt(), nextLong(), nextFloat(), nextDouble(), or nextBoolean() to obtain to a string, byte, short, int, long, float, double, or boolean value. For example,

```
System.out.print("Enter a double value: ");
Scanner input = new Scanner(System.in);
double d = input.nextDouble();
```

Compute Area with console input

```
import java.util.Scanner;                    // Scanner is in the java.util package

public class ComputeAreaWithConsoleInput {
public static void main(String[] args) {
  // Create a Scanner object
  Scanner input = new Scanner(System.in);

  // Prompt the user to enter a radius
  System.out.print(“Enter a number for radius: ”);
    double radius = input.nextDouble();

  // Compute area
   area = radius * radius * 3.14159;

   // Display results
   System.out.println("The area for the circle of radius " +
     radius + " is " + area);
 }
}
```

Variables

- Java has build-in primitives to support boolean, character, integer and floating-point values.

  - boolean either true or false boolean b = true;

  - char 16-bit Unicode 1.1 character char ch = ‘J’;

  - byte 8-bit integer (signed) byte bt = 127;
  - short 16-bit integer (signed) short sh = 32767;
  - int 32-bit integer (signed) int i = 2147483647;
  - long 64-bit integer (signed) long l = 9223372036854775807L;

  - float 32-bit floating-point (IEEE 754-1985) float f = 1.0f;
  - double 64-bit floating-point (IEEE 754-1985) double d = 1.e-1;

Constants

- A named constant is an identifier that represents a permanent value.

```
final datatype CONSTANTNAME = VALUE;


final double PI = 3.14159;

final int SIZE = 3;
```

Flow of Control

- Flow of control is the term used for describing which statements are executed in a program.
- Flow statements are:
  - if – else
  - for
  - switch
  - do – while
  - while
  - block of code – statements group within { and }

Anatomy of a Java Program

- Comments (//, /*…*/, /**…*/)
- Reserved words (public, void, static, class etc.)
- Modifiers (public, static)
- Statements (System.out.println("Welcome to Java!")
- Blocks (Class Block, Method Block)
- Classes (Welcome)
- Methods (At least one method is required)
- The main method

Comments

Three types of comments in Java.

1. *Line comment*: A line comment is preceded by two slashes (//) in a line.
2. *Paragraph comment*: A paragraph comment is enclosed between /* and */ in one or multiple lines.

3. *javadoc comment*: javadoc comments begin with /** and end with */. They are used for documenting classes, data, and methods. They can be extracted using JDK's javadoc command.

- **Reserved words:** Reserved words or keywords are words that have a specific meaning to the compiler and cannot be used for other purposes in the program.

- **Modifiers:** Java uses certain reserved words called modifiers that specify the properties of the data, methods, and classes and how they can be used.

- **Statements:** A statement represents an action or a sequence of actions.