SNS COLLEGE OF TECHNOLOGY

Coimbatore-35.

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

COURSE CODE & NAME : 23CST205 - Object Oriented Programming Using Java

II YEAR/ III SEMESTER

UNIT – II INTRODUCTION TO JAVA

Topic: BASICS OF JAVA PROGRAMMING-VARIABLES

Java Variables

- Variables are containers for storing data values.
- In Java, there are different types of variables, for example:
  - 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
  - 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

Declaring (Creating) Variables

- To create a variable, you must specify the type and assign it a value:
- Syntax

type variableName = value;

- Where type is one of Java's types (such as int or String), and variableName is the name of the variable (such as x or name). The equal sign is used to assign values to the variable.

Declaring (Creating) Variables

- Example
  - Create a variable called name of type String and assign it the value “Robin".
  - Then we use println() to print the name variable:

```
public class Main {
  public static void main(String[] args) {
    String name = " Robin";
    System.out.println(name);
  }
}
```

- Output: Robin

- Example
  - Create a variable called myNum of type int and assign it the value 35:

```
public class Main {
  public static void main(String[] args) {
    int myNum = 35;
    System.out.println(myNum);
  }
}
```

- Output: 35

TYPES OF VARIABLES

```
Eg: public class A
{
   static int m=100;//static variable
   void method()
   {
      int n=90;//local variable
   }
   public static void main(String args[])
   {
      int data=50;//instance variable
   }
}//end of class
```

1) Local Variable

- A variable declared inside the body of the method is called local variable. You can use this variable only within that method and the other methods in the class aren't even aware that the variable exists.
- A local variable cannot be defined with "static" keyword.

2) Instance Variable

- A variable declared inside the class but outside the body of the method, is called an instance variable. It is not declared as static.
- It is called an instance variable because its value is instance-specific and is not shared among instances.

3) Static variable

- A variable that is declared as static is called a static variable. It cannot be local. You can create a single copy of the static variable and share it among all the instances of the class. Memory allocation for static variables happens only once when the class is loaded in the memory

Final Variables

- If you don't want others (or yourself) to overwrite existing values, use the final keyword (this will declare the variable as "final" or "constant", which means unchangeable and read-only):

- Example:

```
public class Main {
  public static void main(String[] args) {
    final int myNum = 25;
    myNum = 30; // will generate an error
    System.out.println(myNum);
  }
}
```

- Output:

```
Main.java:4: error: cannot assign a value to final variable myNum
    myNum = 30;
        ^
1 error
```

Display Variables

- The println() method is often used to display variables.
- To combine both text and a variable, use the + character:
- Example:

```
public class Main {
  public static void main(String[] args) {
    String name = "John";
    System.out.println("Hello " + name);
  }
}
```

- Output: Hello John

Display Variables

- You can also use the + character to add a variable to another variable:
- Example:

```
public class Main {
  public static void main(String[] args) {
    String firstName = "John ";
    String lastName = "David";
    String fullName = firstName + lastName;
    System.out.println(fullName);
  }
}
```

- Output: John David

Declare Many Variables

- To declare more than one variable of the same type, you can use a comma-separated list
- Example

Instead of writing:

```
int x = 5;

int y = 6;

int z = 50;

System.out.println(x + y + z);
```

You can simply write:

```
int x = 5, y = 6, z = 50;

System.out.println(x + y + z);
```

One Value to Multiple Variables

- You can also assign the same value to multiple variables in one line:
- Example

```
int x, y, z;

x = y = z = 50;

System.out.println(x + y + z);
```

Java Identifiers

- All Java variables must be identified with unique names.
- These unique names are called identifiers.
- Identifiers can be short names (like x and y) or more descriptive names (age, sum, totalVolume).
- Note: It is recommended to use descriptive names in order to create understandable and maintainable code
- Example

```
// Good
int minutesPerHour = 60;
// OK, but not so easy to understand what m actually is
int m = 60;
```

Java Identifiers

- The general rules for naming variables are:
  - Names can contain letters, digits, underscores, and dollar signs
  - Names must begin with a letter
  - Names should start with a lowercase letter, and cannot contain whitespace
  - Names can also begin with $ and _ (but we will not use it in this tutorial)
  - Names are case-sensitive ("myVar" and "myvar" are different variables)
  - Reserved words (like Java keywords, such as int or boolean) cannot be used as names

Real-Life Examples

- A program that stores different data about a college student

```
public class Main {
  public static void main(String[] args) {
    // Student data
    String studentName = "John David";
    int studentID = 15;
    int studentAge = 23;
    float studentFee = 75.25f;
    char studentGrade = 'B';
      // Print variables
    System.out.println("Student name: " + studentName);
    System.out.println("Student id: " + studentID);
    System.out.println("Student age: " + studentAge);
    System.out.println("Student fee: " + studentFee);
    System.out.println("Student grade: " + studentGrade);
  }
}
```

Output:

Student name: John David
Student id: 15
Student age: 23
Student fee: 75.25
Student grade: B


THANK
YOU!