CodInClub

Powered By BridgeLabz

Programming Concept Practiced

1. **Java Comments -** Comments are part of code ignored by Java Compiler. It's used to increase the readability of code.

   // => Single Line Comments

   /* */ => Multiline Comments

2. **Java Data Types**

   int => Integer Data Type takes numbers without decimals

   double => Floating Point Numbers with decimals

   char => Single Character wrapped in single quotes ‘’

   String => Text wrapped in double quotes “” are strings

3. **Java Variables -** Allows programmers to store data in the memory of a particular Java Data Types for later use in the Program. Java Variables can be of type int, double, char, String, etc. We will practice the following

   a. Declare Variable of a particular type
   b. Assign Value
   c. Print Value
   d. Change Value assigned to a Variable

4. **Variable Naming -** Selecting a proper variable name is good programming practice.

   a. Single word Variable => For Salary use *salary* word instead of some character *s*
   b. Multi-word Variable => For the Annual Salary variable use camelCase like annualSalary
   c. Wrong Variable Naming => For Flat # 1 use variable flatNumber instead of flat#

```java
// Single Word Variable
int salary = 100000; // Good Programming Practice
int s = 100000; // Bad Programming Practice

// Total Word Variable
int annualSalary = 1000000; // Good Programming Practice
int as = 1000000; // Bad Programming Practice

// Illegal Variable Naming
int flatNumber = 1; // Good Programming Practice
int flat# = 1; // Bad Programming Practice
```

5. **Display Text and Variables -**
    a. Display Text and Variables in a single print statement using the `+` operator is a Good Programming Practice.
    b. Similarly, Multiple Variable and Text can also be displayed
    c. Display Text in a New Line, use `\n`
6. **Arithmetic Operators -** Arithmetic operators are used to perform arithmetic operations such as addition, subtraction, etc. Here's a list of arithmetic operators

Operators	Operations
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Remainder after division (Modulo)

7. **Multi-Variable Together -** Multi Variable in a single line of the same types is doable and acceptable practice

```java
// Multiline Variable for Salary and Annual Salary
int salary = 100000, annualSalary = 1000000;

/*
 * Increment Salary by 10%. Note use of arithmatic operators add (+),
 * multiplication Operator (*) and Division Operator (/)
 */
int increment = salary * 10/100;
salary = salary + increment;

// Displaying Annual Salary Text and Vriable
System.out.println("Annual Salary " + annualSalary);

// Displaying Increment and new Salary using multi text and variable
System.out.println("Increment = " + increment + " New Salary " + salary);

// Displaying Old Salary, Increment and New Salary In Multi Line
System.out.println("Old Salary = " + (salary-increment) + "\nIncrement = " +
increment + "\nNew Salary " + salary);
```

8. **Operator Precedence -** Referred by Acronym for Precedence from Left to Right **PEMDAS** (Parenthesis, Exponents, Multiplication, Division, Addition and Subtraction) or BODMAS (Brackets, Of, Division, Multiplication, Addition and Subtraction)
    a. Rule # 1: Computation is from Left to Right
    b. Rule # 2: Parenthesis have the highest precedence `()`
    c. Rule # 3: Multiplication `*`, Division `\` and Modulus `%` have the highest precedence over all the Arithmetic Operators. These 3 have equal precedence and are dependent on computation from left to right
    d. Rule # 4 Addition and Subtraction - This has the least precedence and is dependent on calculation from left to right

```java
// First: 3 / 2 is computed = 1
// Second: the result 1 is multiplied by 4 = 4
// Third: 9 is added to 4  = 13
// Fourth: 13 is subtracted by 2 = 11
int x = 9 + 3 / 2 * 4 - 2;
System.out.println(x); // Output is 11

// Actually the intent was to compute (9+3) First then divide by 2, the
// result multiply by 4 and finally subtracted by 2 to result in 22
int y = (9 + 3) / 2 * 4 - 2;
System.out.println(y);
```

9. **Type Conversion -** This is converting a value from one data type to another either done implicitly by Java or explicitly by the user.

```java
// We know 1 Litre is 1000 ml so how much is 500 ml in litre
double halfLitre = 500/1000;

// Result is 0.0 this is because 500 and 1000 are both integer and hence when
// devided the value 0.5 is converted to int type and to double implecitely
System.out.println(halfLitre + "l");

// The correct computation is convert 500 and 1000 explecitely to double
halfLitre = (double)500 / (double)1000;
System.out.println(halfLitre + "l"); // Output is 0.5

int value1 = 20 + 45.68; // Compiler Error as double cannot be converted to int
int value2 = 20 + (double) 45.68; // Correction using Explicit Conversion
double value3 = 20 + 45.68; // Will Work Correctly and show accurate result
```

10. **User Input -** If we want the user to give an input and assign that to a variable, we can use the Scanner class to read user input.
    a. We start by importing the Scanner class using `import java.util.Scanner;`
    b. Create an Object of Scanner class using `Scanner input = new Scanner(System.in);`
    c. **Note:** `System.in` used in the above code means we are taking input from the standard input medium or stream i.e. the keyboard.
    d. Take input from the user we use the functions like nextInt(), nextDouble(), etc as in the following code `int age = input.nextInt(); double height = input.nextDouble();`
    e. Finally, its a good practice to close the scanner object

Java
```java
// Writing Java Code to take user contact details as input and display it.
import java.util.Scanner;
public class ContactDetails {
    public static void main(String[] args) {
        // Creating variables to store user contact details
        String name, email, phone;
        int age;
        double height;

        // Creating Scanner object to take input from user
        Scanner input = new Scanner(System.in);

        // Taking user contact details as input
        System.out.print("Enter your name: ");
        name = input.nextLine();
        System.out.print("Enter your email: ");
        email = input.nextLine();
        System.out.print("Enter your phone number: ");
        phone = input.nextLine();
        System.out.print("Enter your age: ");
        age = input.nextInt();
        System.out.print("Enter your height: ");
        height = input.nextDouble();

        // Displaying user contact details in a single line
        System.out.println("User Contact Details:");
        System.out.println("Name: " + name + ", Email: " + email +
                           ", Phone: " + phone +
                           ",\nAge: " + age + ", Height: " + height);
        input.close(); // Closing the Scanner object
    }
}
```

Best Programming Practice

1. All values as variables including Fixed, User Inputs, and Results
2. Avoid Hard Coding of variables wherever possible
3. Proper naming conventions for all variables
   ```
   String name = "Eric";
   double height = input.nextDouble();
   double totalDistance = distanceFromToVia + distanceViaToFinalCity;
   ```
4. Proper Program Name and Class Name
5. Follow proper indentation

1. **Sample Program 1 -** Write a program to display Sam with Roll Number 1, Percent Marks 99.99, and the result 'P' indicates Pass('P') or Fail ('F').

   IMP => Follow Good Programming Practice demonstrated below in all Practice Programs

```java
// Creating Class with name DisplayResult indicating the purpose is to display
// result. Notice the class name is a Noun.
class DisplayResult {
    public static void main(String[] args) {

        // Create a string variable name and assign value Sam
        String name = "Sam";

        // Create a int variable rollNumber and assign value 1
        int rollNumber = 1;

        // Create a double variable percentMarks and assign value 99.99
        double percentMarks = 99.99;

        // Create a char variable result and assign value 'P' for pass
        char result = 'P';

        // Display the result
        System.out.println("Displaying Result:\n" +name+ " with Roll Number " +
                           rollNumber+ " has Scored " +percentMarks+
                           "% Marks and Result is " +result);
    }
}
```

2. **Sample Program 2 -** Eric Travels from Chennai to Bangalore via Vellore. From Chennai to Vellore distance is 156.6 km and the time taken is 4 Hours and 4 Mins and from Vellore to Bangalore is 211.8 km and will take 4 Hours and 25 Mins. Compute the total distance and total time from Chennai to Bangalore

```java
// Create TravelComputation Class to compute the Distance and Travel Time

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

      // Create a variable name to indicate the person traveling
      String name = "Eric";

      // Create a variable fromCity, viaCity and toCity to indicate the city
      // from city, via city and to city the person is travelling
      String fromCity = "Chennai", viaCity = "Velore", toCity = "Bangalore";

      // Create a variable distanceFromToVia to indicate the distance
      // between the fromCity to viaCity
      double distanceFromToVia = 156.6;

      // Create a variable timeFromToVia to indicate the time taken to
      // travel from fromCity to viaCity in minutes
      int timeFromToVia = 4 * 60 + 4;

      // Create a variable distanceViaToFinalCity to indicate the distance
      // between the viaCity to toCity
      double distanceViaToFinalCity = 211.8;

      // Create a variable timeViaToFinalCity to indicate the time taken to
      // travel from viaCity to toCity in minutes
      int timeViaToFinalCity = 4 * 60 + 25;

      // Create a variable totalDistance to indicate the total distance
      // between the fromCity to toCity
      double totalDistance = distanceFromToVia + distanceViaToFinalCity;

      // Create a variable totalTime to indicate the total time taken to
      // travel from fromCity to toCity in minutes
      int totalTime = timeFromToVia + timeViaToFinalCity;
```

```
        // Print the travel details
        System.out.println("The Total Distance travelled by " + name + " from " +
                           fromCity + " to " + toCity + " via " + viaCity +
                           " is " + totalDistance + " km and " +
                           "the Total Time taken is " + totalTime + " minutes");
    }
}
```