

Title: Introduction To Basic Java Program

Objectives:

- Write, compile and run Java programs.
- Declare and define classes and objects.
- Declare fields, methods and constructors.
- Read input from commandline.

Theory:

- **Writing, compiling and running java programs:** To compile a java program we need to install the JDK (Java Development Kit) and add the java binaries to the path. We create a file with .java extension. We use the command `javac MyFile.java` to compile the java file to a `MyFile.class` file and use `java MyFile` to run the code.
- **Declare and define classes and objects:** Java is an object oriented language. Every java file must have a public class with a main method which is the entry point. By convention the Entry class has the same name as the java file. We can define classes by the `class` keyword.

```
class MyClass{
    int a;
}
```

Objects can be instantiated by using the new keyword.

```
class Main{
    public static void main(String args[]){
        MyClass c = new MyClass();
    }
}
```

- **Declare fields, methods and constructors:** Fields and methods are analogous to variables and functions in procedural programming, they are simply variables and functions encapsulated in an object. A constructor is special method that runs whenever an object is instantiated. Fields and methods can be private (Not visible outside of class) or public (visible outside of class). An example code with field method and constructor:

```
class classRoom{
    public int studentNum;
    public static int roomCount=0;
    public void addStudent(){
        studentNum++;
    }
    classRoom(int num){
        studentNum=num;
    }
}
```

- **Read input from commandline:** Command line arguments are captured by the `String args[]` parameter in the main method. The arguments can be accessed by using the args array.

Program 1: Hello World Program

```
public class HelloWorld{
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

Output: Hello World

Program 2: Program to model a Motorbike

```
public class Motorbike {
    int speed;
    String model;

    public void accelerate() {
        this.speed += 10;
    }
    public void stop(){
        this.speed = 0;
    }
    public void printSpeed(){
        System.out.println(model + " is travelling at " + speed);
    }
    public static void main(String[] args) {
        Motorbike honda = new Motorbike();
        honda.model = "Hero";
        honda.speed = 100;
        Motorbike bajaj = new Motorbike();
        bajaj.model = "pulsar";
        bajaj.speed = 140;
        honda.accelerate();
        honda.printSpeed();
        bajaj.stop();
        bajaj.printSpeed();
    }
}
```

Output: Hero is travelling at 110 pulsar is travelling at 0

Program 3: Program to read integer and display

```
import java.io.*;

public class PrintNumber {

    public int getInt() {
        int num = 0;
        System.out.println("Enter a number");
        String line = "";
        try {
            BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
            line = in.readLine();
            num = Integer.parseInt(line);
        } catch (Exception e) {
```

```

        System.err.println("IO error" + e);
    }
    return num;
}

public static void main(String[] args) {
    PrintNumber p = new PrintNumber();
    int a = p.getInt();
    int b = p.getInt();

    System.out.println("Sum = " + (a + b));
}
}

```

Output: Enter a number 4 Enter a number 4 Sum = 8

Program 4: Program to save Data for Bank

```

import java.util.Random;
import java.util.Scanner;
import java.util.Date;
public class Bank {
    String id;
    float balance;
    int transactionCount;
    String name;
    public Bank(){
        Random rand = new Random();
        Date date = new Date();
        this.id = String.valueOf(rand.nextInt(999999)) + String.valueOf(date.getTime());
        this.transactionCount = 0;
    }
    public void getDetails(){
        Scanner in = new Scanner(System.in);
        String line = "";
        System.out.println("Enter Name");
        line = in.nextLine();
        name = line;
        System.out.println("Enter balance");
        line = in.nextLine();
        try{
            balance = Float.parseFloat(line);
        }catch(Exception e){
            System.out.println("Invalid balance");
            balance = 0;
        }finally{
            in.close();
        }
    }
    public void withdraw(float amt){
        if(amt < balance && amt > 0){
            balance -= amt;
            transactionCount++;
        }
    }
}

```

```

    }
}
public void deposit(float amt){
    if(amt >0){
        balance+=amt;
        transactionCount++;
    }
}
public void display(){
    System.out.println("Id = " + id);
    System.out.println("Name = " + name);
    System.out.println("Balance = " + balance);
    System.out.println("Transaction Count = " + transactionCount);
}
public static void main(String[] args) {
    Bank myBank = new Bank();
    myBank.getDetails();
    myBank.deposit(3000);
    myBank.withdraw(5000);
    myBank.display();
}
}

```

Output: Enter a number 4 Enter a number 5 Sum = 9

Program 5: Program to count instances of a class:

```

public class ClassCount {
    static int count = 0;
    ClassCount(){
        ClassCount.count++;
    }
    public static void main(String[] args) {
        ClassCount c1 = new ClassCount();
        ClassCount c2 = new ClassCount();
        ClassCount c3 = new ClassCount();
        System.out.println(ClassCount.count);
    }
}

```

Output: 3

Program 6: Program that reads two numbers from the command line, the number of hours worked by an employee and their base pay rate. Add warning messages if the pay rate is less than the minimum wage (12.5 an hour) or if the employee worked more than the number of hours in a week.

```

public class PayRate {
    public static void main(String[] args) {
        if (args.length == 2) {
            try {

                float hours = Float.valueOf(args[0]);
                float rate = Float.valueOf(args[1]);

```

```

        float pay = hours * rate;
        if (rate < 12.5) {
            System.out.println("Less than minimum rate");
        }
        if (hours > 7.24) {
            System.out.println("Over worked");
        }
        System.out.println("Pay = " + pay);
    } catch (Exception e) {
        System.out.println("Error parsing arguments");
    }
} else {
    System.out.println("Provide 2 arguments");
}
}
}

```

Output:

```

~/coding/java/lab1: java PayRate 10 50
Over worked
Pay = 500.0

```

Conclusion: Java programs were written such that we learnt about java compilation, constructors, fields, methods and objects and Command Line Arguments.