

Q3. First Java Program.

input/output, debugging, datatype.

Date _____
Page _____

* Structure of Java file.

The filename is Main.java; the file should contain a class and name of the class is 'main'.

Ex.

Main.java ↓
class Main {
}

- If you created a class that name 1st letter should be capital bcz its a good practice.
- If the name of the file is Main.java then main class in the file getting public.
- If you want to showcase any of the classes to some people then you can use "private" if you want it access should be for all the member, so then it will be 'public'.
- The public & private is an **access specifier**. **Modifier**.
- **Public** Mean, the class can access from anywhere.
- In Java it has main function, from that we start to do code. the function is block of code.
- **Function** : the collection of code you use again and again.
- **Java main function**:

public static void main (String args[]) {}

- **public**: It mean the class can access from anywhere.
- **class**: Class is main group of properties & methods.
- **main()**: The main function is a entry point of Java program.
- **static**: We want run the main function without creating a object of class Main. That's why we used static.

→ void : void is a return type of function

→ command line argument : string args []

```
public class Demo {  
    public static void main ( ) {  
    }  
}
```

// This is how we can structure the Java Program.

→ package : in simple, the folder that Java file lies.

→ System.out.println : the system is a class that contains variable out, out is a type of print stream, and out actually has a method println () .

→ input in.java

→ scanner : which can pass primitive type

→ system.in : it's a standard input stream. typically this stream corresponds to keyboard input.

scanner input = new scanner (System.in) ;

→ taking input, bcz it will contain all function provided by scanner class.

→ Input.nextInt () ; "it will gives a integer value from user."

* Primitive data type.

↳ Integer.

→ if you want to store Roll No. of student

"int roll = 64;"

→ if you want to store PI value

"float pi = 3.14;"

→ Primitive : Any data type we cannot break even further

→ ex.

String name = "Bhuuj";

→ we can break string in any section it means we can modify it
so that's why string is not primitive data type.

→ "char letter = 'g';"

→ use double for large decimal Number

"double largeDecimal = 456708.93734;"

→ long data type use for large integer.

"long integerType = 8456891076734L;"

→ To write / assign true / false we used boolean data type.

"bool check = true;"

- If you have already had 'int' why use long and you have float why use double?
- we would look into the maximum size of store in one data type.

int have 4 byte
 float have 4 byte
 double have 8 byte
 long also 8 byte

} size of datatype.

- Why we are adding 'f' and 'L' when you declare variable using float & Long data type.
- Basically all the decimal value by default that type is "double", if you want to store that into float basically we have to add "f".
- Taking input using primitive data type.

public class Inputs {

 public static void main (String args[]) {

 Scanner input = new Scanner (System.in);

 int Roll.NO = input.nextInt();

 System.out.print (Roll.NO); ... it give integer input

}

- Literals : in primitive data type is syntactical representation of constant and the bool, numeric.

e.g.

int a = (10) → Literals

↑

identity (reference variable).

→ String input:

String name = input.nextLine();

→ Float input:

float mark = input.nextFloat();

→ Sum of two number.

```
import java.util.Scanner;  
public class Sum {  
    public static void main (String args[]){  
        Scanner input = new Scanner (System.in);  
        System.out.println ("Enter the value of a&b:");  
        int a = input.nextInt();  
        int b = input.nextInt();  
        int c = a+b;  
        System.out.println ("Addition : "+c);  
    }  
}
```

→ Type casting.

1) Type should be compatible.

2) whatever on left hand side / destination should be greater than source right hand side.

→ if one type of data is assign to another type of variable then automatic type conversion will take place if following condition met.

→ casting.

How we can convert type int into float?

"int num = (int) (64.56f);"

- compressing a bigger number in smaller type explicitly. called type casting.
- automatic type promotion in expression.

int a = 257;

byte b = (byte) (a); // 257 % 256 = 1

- whenever we do byte evaluation it automatically converted into 'int': its called type promotion.

eg.

(a + int) num = 210; // 210

S.O.P.(num); // 210

- Java follows unicode.

- Rule for type promotion.

- 1) All byte, short, character value are promoted to integer
- 2) If any one of the operand is long the whole operation promoted to long.

eg.

S.O.P(5 * 3.1);

(if will convert in float.)

→ Assignment

Ques: 01 write a program to print whether a number is even or odd.

→ import java.util.Scanner;

public class OddEven {

 public static void main (String args []) {

 Scanner input = new Scanner (System.in);

 System.out.println ("Enter any number:");

 int num = input.nextInt ();

 if (num % 2 == 0) {

 System.out.println ("Even");

 } else {

 System.out.println ("Odd");

}

}

Ques: 02 take name as input and print a greeting message for that particular name.

→ import java.util.Scanner;

public class greeting {

 public static void name (String args []) {

 Scanner input = new Scanner (System.in);

 System.out.println ("Enter any name:");

 String name = input.nextLine ();

 System.out.println ("Hello: " + name);

}

}

Ques: 03

write a program to input principal, time and rate from the user and find simple interest.

```
→ import java.util.Scanner;
public class SimpleInterest {
    public static void main (String args[]) {
        Scanner input = new Scanner (System.in);
        System.out.println ("Enter the value of principal:");
        int principal = input.nextInt ();
        System.out.println ("Enter the value of time:");
        int time = input.nextInt ();
        System.out.println ("Enter the value of rate:");
        int rate = input.nextInt ();
        int SI = principal * time * rate;
        System.out.println ("Simple interest: " + SI);
    }
}
```

Ques: 04

take input of two number and operators (+,-,*,/) and calculate the value.

```
→ import java.util.Scanner;
public class Calculator {
    public static void main (String args[]) {
        Scanner input = new Scanner (System.in);
        System.out.println ("Enter the value of two number:");
        int a = input.nextInt ();
        int b = input.nextInt ();
        System.out.println ("use operators from +, -, *, /");
        int operator = input.nextInt ();
    }
}
```

```
if (operator == '+') {
    int sum = a+b;
    System.out.println ("Addition :" + sum);
} else if (operator == '-') {
    int sub = a-b;
    System.out.println ("Subtraction :" + sub);
} else if (operator == '*') {
    int mul = a*b;
    System.out.println ("Multiplication :" + mul);
} else if (operator == '/') {
    int div = a/b;
    System.out.println ("Division :" + div);
} else {
    System.out.println ("Invalid operator");
}
}
```

Ques: 05 Take two numbers as input and print the largest number.

```
→ import java.util.Scanner;
public class Max {
    public static void main (String args []) {
        Scanner input = new Scanner (System.in);
        System.out.println ("Enter value of two numbers:");
        int num1 = input.nextInt();
        int num2 = input.nextInt();
        if (num1 > num2) {
            B.o.p ("num1 is greater");
            S.o.p ("num2 is greater");
        }
    }
}
```

Ques: 06. Input currency in Rupees and output in USD.

```

→ import java.util.Scanner;
public class Currency {
    public static void main (String args[]) {
        Scanner input = new Scanner (System.in);
        System.out.println ("Enter the value in ₹:");
        float rupees = input.nextFloat();
        float USD = rupees / 82.89;
        System.out.println ("Currency in " + rupees + " convert into USD" + USD);
    }
}

```

Ques: 07 To calculate Fibonacci series up to n number.

```

→ import java.util.Scanner;
public class Fibonacci {
    public static void main (String args[]) {
        Scanner input = new Scanner (System.in);
        System.out.println ("Enter length of series:");
        int n = input.nextInt();
        int a = 0; // private
        int b = 1; // private
        int c = 0; // private
        int i = 0; // private
        System.out.print (a + ", " + b + ",");
        while (i < n - 2) {
            c = a + b;
            a = b;
            b = c;
            i++;
            System.out.print (", " + c);
        }
    }
}

```

```
c=a+b;  
s.op(c+",");  
a=b;  
b=c;  
i++;  
}  
}  
}  
}
```

Ques: 08. To find whether given string is palindrome.

```
→ import java.util.Scanner;  
public class palindrome {  
    public static void main (String args []){  
        Scanner input = new Scanner (System.in);  
        System.out.println ("Enter any string : ");  
        String name = input.nextLine();  
        String REV="";  
        for (int i= name.length()-1 ; i>0 ; i--) {  
            Rev= Rev + name.charAt(i);  
        }  
        if (name.equals (REV)) {  
            s.o.p ("palindrome");  
        } else {  
            s.o.p ("Not palindrome");  
        }  
    }  
}
```

Ques: Q9 To find Armstrong number between two given numbers.

```
→ import java.util.Scanner;  
public class Armstrong {  
    public static void main(String args[]) {  
        int start = 100;  
        int end = 1000;  
        for (int i = start; i <= end; i++) {  
            int number = i;  
            int sum = 0;  
            int digitCount = String.valueOf(number).length();  
            while (number > 0) {  
                int digit = number % 10;  
                sum = sum + Math.pow(digit, digitCount);  
                number = number / 10;  
            }  
            if (sum == i) {  
                System.out.println(i + " is an Armstrong.");  
            }  
        }  
    }  
}
```