

12/12/23

1. ParseInt :

```
class parseInt {  
    public static void main (String args[]) {  
        int a = Integer.parseInt (args[0]);  
        System.out.print (a);  
    }  
}
```

2. Scanner :

```
import java.util.*;  
class scanner {  
    public static void main (String args[]) {  
        Scanner n = new Scanner (System.in);  
        int a = n.nextInt();  
        System.out.print (a);  
    }  
}
```

3. Factorial :

```
import java.util.*;  
class fact {  
    public static void main (String args[]) {  
        Scanner sc = new Scanner (System.in);  
        int n = sc.nextInt();  
        int f = 1;  
        for (int i = 1; i <= n; i++) f *= i;  
        System.out.print (f);  
    }  
}
```

4. Pallindrome:

```

import java.util.*;
class pallindrome {
    public static void main (String args []) {
        Scanner sc = new Scanner (System.in);
        String s = sc.nextLine();
        String s1 = sc.nextLine();
        for (int i = 0; i < s.length(); i++) {
            s1 += s.charAt(i);
        }
        if (s.equals(s1)) {
            System.out.print("It is a pallindrome");
        }
        else {
            System.out.print("It is not a pallindrome");
        }
    }
}
    
```

5. Sum of digits

```
import java.util.*;  
class sum {  
    public static void main(String args[]) {  
        Scanner sc = new Scanner(System.in);  
        String num = sc.nextLine();  
        int sum = 0, i;  
        for (i = 0; i < num.length(); i++) {  
            sum += Integer.parseInt(String.valueOf(num.charAt(i)));  
        }  
        System.out.print(sum);  
    }  
}
```

6. Arrays 1D:

```
import java.util.*;  
class array {  
    public static void main(String args[]) {  
        System.out.println("Enter the no. of elements");  
        Scanner sc = new Scanner(System.in);  
        int n = sc.nextInt();  
        int A[] = new int[n];  
        System.out.print("Enter elements");  
        for (int i = 0; i < n; i++) {  
            A[i] = sc.nextInt();  
        }  
        for (int i = 0; i < n; i++) {  
            System.out.print(A[i]);  
        }  
    }  
}
```

7. Type conversion:

class type {

public static void main (String args[]) {

int a = 6;

float b = a;

System.out.println ("Implicit type conversion : \n" + b);

double c = 0.3;

int d = (int) c;

System.out.println ("Explicit type conversion : \n" + d);

}

}

8. Quadratic:

import java.util.*;

class quad {

public static void main (String args[]) {

System.out.println ("Nirat Dagan IBM22CS80");

Scanner n = new Scanner (System.in);

int a, b, c;

double x1, x2, ~~z~~ d;

System.out.print ("Enter coefficients a, b, c : ");

a = n.nextInt();

b = n.nextInt();

c = n.nextInt();

while (a == 0) {

System.out.print ("It is not a quadratic eq");

}

d = b * b - 4 * a * c;


```

if (d == 0) {
    r1 = (-b) / (2 * a);
    System.out.println("Roots are real and equal");
    System.out.println("Root 1 = Root 2 = " + r1);
}
else if (d > 0) {
    r1 = ((-b) + (Math.sqrt(d))) / (double)(2 * a);
    r2 = ((-b) - (Math.sqrt(d))) / (double)(2 * a);
    System.out.println("Roots are real and distinct");
    System.out.println("Root 1 = " + r1 + " Root 2 = " + r2);
}
else {
    System.out.println("Roots are imaginary");
    r1 = (-b) / (2 * a);
    r2 = Math.sqrt(-d) / (2 * a);
    System.out.println("Root 1 = " + r1 + " + i " + r2);
    System.out.println("Root 2 = " + r1 + " - i " + r2);
}
}
}
}

```

Output

Niket Dugar ~~IBM22~~ IBM22 CS180

Case 1 Enter coefficients of a, b, c: 1 2 1
Roots are real and equal
Root 1 = Root 2 = -1.0

Case 2 Enter coefficients of a, b, c: 0 1 1
~~It~~ It is not a quadratic equation

Case 3 Enter coefficients of a, b, c: 1 1 2
Roots are imaginary
Root 1 = 0.0 + i 1.322875
Root 2 = 0.0 - i 1.322875

Case 4 Enter coefficients of a, b, c : 1 -3 2

Roots are real and distinct

Root 1 = 2.0 Root 2 = 1.0

12/12