

JAVA Programming

- 1.Data Type (Boolean, char, short, int, float, double)
- 2.Format Specifier (Printing using format specifier)
- 3.Take input from user then print the value
- 4.Print a string
- 5.Print a string from the user
- 6.Print a double number from the user
- 7.Arithmetic operator
- 8.Arithmetic operator from the user(+, -, *, /)
- 9.Assignment operator (+=)
- 10.Area of a triangle from the user
- 11.Area of a circle from the user
12. Celsius to farenheight from the user
13. Unary operator (x+, +x)
- 14.1 Increment and Decrement
- 14.2 Increment and Decrement
- 14.3 Increment and Decrement
15. Positive or Negative
16. Even or Odd
17. Vowel or Consonant
18. Small or Capital letter
19. Digit using switch
20. Condition (? :)
21. Bitwise and, or, not
22. Binary operator (>>, <<)
23. Math Class(Different method of math class)
24. For loop

- 25. While loop
- 26. do while loop
- 27.1 Break and Continue Statement
- 27.2 Break and Continue Statement
- 27.3 Break and Continue Statement
- 28. Print sum of all the number from 1 to 10
- 29. Print sum of all the number from m to n
- 30. Print sum of all the even number from m to n
- 31. Print sum of $1+2+3+\dots+n$
- 32. Print sum of $2+4+6+\dots+n$
- 33. Print sum of $1+3+5+\dots+n$
- 34. Print sum of $1.5+2.5+3.5+\dots+n$
- 35. Print sum of $1^2+2^2+3^2+\dots+n$
- 36. Print sum of $1*2*3*\dots*n$
- 37. Factorial printing
- 38.1 Multiplication table
- 38.2 Multiplication table
- 39. Prime or not
- 40. Prime number from m to n
- 41. Fibonacci Number Print
- 42. Sum of digit Print
- 43. Reverse a number and print it
- 44. Palindrome number print
- 45. Armstrong number print
- 46.1 Pattern print
- 46.2 Pattern print
- 47. Array sum and average

- 48.1 Array sum and average from the user
- 48.2 Array sum and average from the user
- 49. Array-Finding maximum and minimum from the user
- 50. Printing string
- 51. Print string using for each loop
- 52. Print a simple matrix
- 53. Printing a matrix and its sum
- 54. Printing the sum of diagonal, upper and lower triangle element.
- 55.1 Printing Array List
- 55.2 Printing Array List
- 56. Array Sorting
- 57. Printing Array List Using Normal/foreach/Iterator method
- 58. Printing Array List Using Normal/foreach/Iterator method.
- 59. Array List
- 60. Sorting Array List
- 61.1 String print
- 61.2 String print
- 61.3 String print
- 61.4 String part
- 62. String Buffer
- 63. Checking palindrome or not
- 64.1 StringBuilder
- 64.2 StringBuilder
- 65. Number Conversion (Decimal, Binary)
- 66. Printing Date
- 67. Time Printing
- 68. Random Number

- 69. Class creation
- 70. Introducing method
- 71. Parametrized method
- 72. Constructor
- 73. Constructor-2
- 74. Overloading constructor
- 75. Returning value
- 76. Static keyword
- 77.1 Static variable-1
- 77.2 Static variable-2
- 78. Static variable-3
- 79. Static method
- 80.1 Static block
- 80.2 Static block-2
- 81. Types of variables (Static, Instance, Local)
- 82. Display volume
- 83. Instance variable hiding
- 84. Math class (pow, log, max, min, ceil, floor, sqrt, pi)
- 85. Method overloading
- 86. Automatic type conversion in java
- 87. Debugging java code
- 88. Call by value
- 89. Call by reference
- 90. Variable length arguments
- 91 Factorial using recursion
- 92. Factorial
- 93. Encapsulation

- 94. Setter and getter method.
- 95. Inheritance
- 96. Inheriting private number
- 97. Instance of operation
- 98. Types of inheritance
- 99. Method overriding
- 100. Difference between overloading and overriding.
- 101. Super keyword-1
- 102. Super keyword-2
- 103. Super class-3
- 104. Super class-4
- 105. This keyword-1
- 106. This keyword-2
- 107. Final keyword-2
- 108. Final keyword-3
- 109. Polymorphism-1
- 110. Polymorphism-2
- 111. Polymorphism-3
- 112. Abstraction-1
- 113. Abstraction-2
- 114. Problem solving
- 115. Interface
- 116. Why interface support multiple inheritance
- 117. Interface vs Abstract
- 118. Package-1
- 119. Package-2
- 120. Access modifier

- 121. Type casting
- 122. Anonymous class
- 123. Exception handling
- 124. Exception handling
- 125. Exception handling problem solve
- 126. Decimal format
- 127. `toString` method
- 128. String comparison
- 129. Linked list-1
- 130. Linked list-2
- 131. Creating a student list
- 132. `HashMap`
- 133. `HashSet`
- 134. How to create a file and folder
- 135. How to write within a file
- 136. How to read a file
- 137.1 Remove an element from an array
- 137.2 Remove an element from an array using method/function
- 138. UML-1
- 139. UML-2
- 140. UML-3
- 141. UML-4
- 142. UML-5
- 143. UML-6
- 144. UML-7
- 145. Lab Task-1 solve
- 146. Lab Task-2 solve

- 147. Creating a simple food order app.
- 148. Polymorphism
- 149. Method overloading
- 150. Method overriding
- 151. Super Keyword part-1
- 152. Super Keyword part-2
- 153. Lab Task-3 solve
- 154. Lab Task-4 solve
- 155. Lab Task-5 solve
- 156. Instance Initializer
- 157. Aggregation in Java
- 158. Setter and Getter method using Constructor

1. Data Type(Boolean, char, short, int, float, double)

```
package datatype;
public class DataType
{
    public static void main(String[] args)
    {
        boolean b = true;
        char c = 'c';
        short s = 32677;
        int i = 126587;
        float f = 12.5f;
        double d = 12.5;

        System.out.println("b = " +b);
        System.out.println("c = " +c);
        System.out.println("s = " +s);
        System.out.println("i = " +i);
        System.out.println("f = " +f);
        System.out.println("d = " +d);
    }
}
```

Output:

```
b = true
c = c
s = 32677
i = 126587
f = 12.5
d = 12.5
```

2. Format Specifier(Printing using format specifier)

```
package datatype;  
public class DataType  
{  
    public static void main(String[] args)  
    {  
        boolean b = true;  
        char c = 'c';  
        short s = 32677;  
        int i = 126587;  
        float f = 12.5f;  
        double d = 12.5;  
  
        System.out.printf("b = %b\n", b);  
        System.out.printf("c = %c\n", c);  
        System.out.printf("s = %d\n", s);  
        System.out.printf("i = %d\n", i);  
        System.out.printf("f = %.2f\n", f);  
        System.out.printf("d = %.2f\n", d);  
    }  
}
```

Output:

```
b = true  
c = c  
s = 32677  
i = 126587  
f = 12.50  
d = 12.50
```

3. Take input from user then print the value

```
package javabeginner;  
import java.util.Scanner;  
/*
```

Akany scanner class er location ta boly diyechi.

Orthat scanner ta holo java dot util package er ontorgoto akta class.

```
*/
```

```
public class UserInput  
{
```

```
    public static void main(String[] args)
```

```
    {
```

```
        int num;
```

```
        System.out.print("Enter any number: ");
```

```
        Scanner input = new Scanner(System.in);
```

```
        /*
```

Java te input neyar jonno ay scanner class use kora hoy.

Akany input holo scanner class er akta variable.

Input er poriborty j kono nam use kora jaby.

```
    */
```

```
        num = input.nextInt();
```

```
        System.out.println("The number is: "+num);
```

```
    }
```

```
}
```

Output:

```
Enter any number: 12
```

```
The number is: 12
```

4. Print a string

```
package javabeginner;

public class UserInput
{
    public static void main(String[] args)
    {
        String name;
        name = "Golam Kibria";
        System.out.println("Name is: "+name);
    }
}
```

Output: Name is: Golma Kibria

5. Print a string from the user

```
package javabeginner;

import java.util.Scanner;

public class UserInput
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        String name;
        System.out.print("Enter your name: ");
        name = input.nextLine();
        /*next er poriborty nextLine use korbo. Atai better way*/
        System.out.println("Welcome "+name);
    }
}
```

Output:

```
Enter your name: Golam kibria
Welcome Golam Kibria
```

6. Print a double number from the user

```
package javabeginner;

import java.util.Scanner;

public class UserInput
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        double num;
        System.out.print("Enter any double numbr: ");
        num = input.nextDouble();
        System.out.println("The number is: "+num);
    }
}
```

Output:

```
Enter any double numbr: 12.34
```

```
The number is: 12.34
```

7. Arithmetic operator

```
package javabeginner;

public class ArithmeticOperatore
{
    public static void main(String[] args)
    {
        int num1, num2, result;
        num1 = 10;
        num2 = 5;

        result = num1+num2;
        System.out.println("Sum: "+result);

        result = num1-num2;
        System.out.println("Sub: "+result);

        result = num1*num2;
        System.out.println("Mul: "+result);

        result = num1/num2;
        System.out.println("Div: "+result);

        result = num1%num2;
        System.out.println("Remainder: "+result);
    }
}
```

Output:

Sum: 15

Sub: 5

Mul: 50

Div: 2

Remainder: 0

8. Arithmetic operator from the user(+, -, *, /)

```
package javabeginner;

import java.util.Scanner;

public class ArithmeticOperatore
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int num1, num2, result;
        System.out.print("Enter first number: ");
        num1 = input.nextInt();
        System.out.print("Enter second number: ");
        num2 = input.nextInt();

        result = num1+num2;
        System.out.println("Sum: "+result);

        result = num1-num2;
        System.out.println("Sub: "+result);

        result = num1*num2;
        System.out.println("Mul: "+result);

        double result2 = (double)num1/num2;
        System.out.println("Div: "+result2);
```

```
    result = num1%num2;  
    System.out.println("Remainder: "+result);  
}  
}
```

Output:

Enter first number: 10

Enter second number: 3

Sum: 13

Sub: 7

Mul: 30

Div: 3.333333333333335

Remainder: 1

9. Assignment operator(+=)

```
package javabeginner;

public class AssignmentOperator
{
    public static void main(String[] args)
    {
        int x = 3, y = 2;

        x+=y; //x = x+y = 5
        System.out.println("x = "+x);
        x-=y; //x = x-y = 3
        System.out.println("x = "+x);
        x*=y; //x = x*y = 6
        System.out.println("x = "+x);
        x/=y; //x = x/y = 3
        System.out.println("x = "+x);
        x%=y; //x = x%y = 1
        System.out.println("x = "+x);
    }
}
```

Output:

```
x = 5
x = 3
x = 6
x = 3
x = 1
/*From the user ayta try korlei parbo*/
```

```
10.Area of a triangle from the user
package javabeginner;

import java.util.Scanner;

public class TriangleArea
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);

        double base, height, area;
        System.out.print("Enter the base: ");
        base = input.nextDouble();
        System.out.print("Enter the height: ");
        height = input.nextDouble();

        area = 0.5 * base * height;

        System.out.println("The area of triangle is: "+area);
    }
}
```

Output:

```
Enter the base: 3
Enter the height: 10
The area of triangle is: 15.0
```

```
11.Area of a circle from the user  
package javabeginner;  
  
import java.util.Scanner;  
  
public class RadiusArea  
{  
    public static void main(String[] args)  
    {  
        Scanner input = new Scanner(System.in);  
  
        double r, area;  
        System.out.print("Enter the radius: ");  
        r = input.nextDouble();  
  
        area = 3.1416 * r * r;  
  
        System.out.println("The area of circle is: "+area);  
    }  
}
```

Output:

```
Enter the radius: 3  
The area of circle is: 28.2744
```

12. Celcious to farenheight from the user

```
package javabeginner;

import java.util.Scanner;

public class RadiusArea
{
    public static void main(String[] args)
    {
        //f = 9/5*c+32, c = 5/9*(f-32)
        Scanner input = new Scanner(System.in);

        double c, f;
        System.out.print("Enter the celcious temp: ");
        c = input.nextDouble();

        f = 1.8 * c + 32;
        System.out.println("The farenheight temp is: "+f);

    }
}
```

Output:

```
Enter the celcious temp: 36
The farenheight temp is: 96.8
```

13. Unary operator(x+, +x)

```
package javabeginner;
```

```
public class UnaryOperator
{
    public static void main(String[] args) {
        int x, result;
        x = 10;

        result = +x;
        System.out.println("x = "+result);
        result = -x;
        System.out.println("x = "+result);
    }
}
```

Output:

```
x = 10
```

```
x = -10
```

14.1 Increment and Decrement

```
package javabeginner;
```

```
public class IncDec
{
    public static void main(String[] args) {
        int x, y;
        x = 10;

        y = x++; //y = 10
        System.out.println("y = " + y);

        //Protom bar barby na. Kintu next a abar kono x pele saty saty man 1 bery jaby.
        y = x; //y = 11
        System.out.println("y = " + y);
    }
}
```

Output:

```
y = 10
```

```
y = 11
```

14.2 Increment and Decrement

```
package javabeginner;

public class IncDec
{
    public static void main(String[] args) {
        int x, y;
        x = 10;

        y = ++x; //y = 11^
        System.out.println("y = " + y);

        //Protom bar barby na. Kintu next a abar kono x pele saty saty man 1 bery jaby.
        y = x; //y = 11
        System.out.println("y = " + y);
    }
}
```

Output:

```
y = 11
y = 11
```

14.3 Increment and Decrement

```
package javabeginner;

public class IncDec
{
    public static void main(String[] args) {
        int x, y;
        x = 10;

        y = --x; //y = 9
        System.out.println("y = "+y);
        y = x--; //y = 9
        System.out.println("y = "+y);
        y = ++x; //y = 8+1 = 9
        System.out.println("y = "+y);
        y = x++; //y = 9
        System.out.println("y = "+y);

    }
}
```

Output:

```
y = 9
y = 9
y = 9
y = 9
```

15. Positive or Negative

```
package javabeginner;
import java.util.Scanner;
public class PositiveNegative
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);

        int num;
        System.out.print("Enter an integer number: ");
        num = input.nextInt();

        if(num>0)
        {
            System.out.println("Positive");
        }
        else if(num<0)
        {
            System.out.println("Negative");
        }
        else
        {
            System.out.println("Zero");
        }
    }
}

Enter an integer number: 5
Positive
```

16. Even or Odd

```
import java.util.Scanner;

public class EvenOdd
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);

        int num;
        System.out.print("Enter a number: ");
        num = input.nextInt();

        if(num%2 == 0){
            System.out.println("Even");
        }else{
            System.out.println("Odd");
        }
    }
}
```

Output:

Enter a number: 6

Even

17. Vowel or Consonant

```
package javabeginner;

import java.util.Scanner;

public class VowelConsonant
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);

        char ch;
        System.out.print("Enter a character: ");
        ch = input.next().charAt(0);

        if(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u'){
            System.out.println("Vowel");
        }else{
            System.out.println("Consonant");
        }
    }
}
```

Output:

```
Enter a character: q
```

```
Consonant
```

18. Small or Capital letter

```
package javabeginner;

import java.util.Scanner;

public class CapitalSmall
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);

        char ch;
        System.out.print("Enter a character: ");
        ch = input.next().charAt(0);

        if(ch >= 'a' && ch <= 'z'){
            System.out.println("Small letter");
        }
        else if(ch >= 'A' && ch <= 'Z'){
            System.out.println("Capital letter");
        }
        else{
            System.out.println("Not a letter");
        }
    }
}
```

Output:

```
Enter a character: Q
```

```
Capital letter
```

19. Digit using switch

```
package javabeginner;

import java.util.Scanner;

public class SwitchUser
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int digit;
        System.out.println("Enter a digit: ");
        digit = input.nextInt();

        switch(digit)
        {
            case 0:
                System.out.println("zero");
                break;
            case 1:
                System.out.println("one");
                break;
            case 2:
                System.out.println("two");
                break;
            case 3:
                System.out.println("three");
                break;
        }
    }
}
```

```
case 4:  
    System.out.println("four");  
    break;  
  
case 5:  
    System.out.println("five");  
    break;  
  
case 6:  
    System.out.println("six");  
    break;  
  
case 7:  
    System.out.println("seven");  
    break;  
  
case 8:  
    System.out.println("eight");  
    break;  
  
case 9:  
    System.out.println("nine");  
    break;  
  
default:  
    System.out.println("Not a digit");  
    break;  
}  
  
}  
  
Output:  
Enter a digit:  
3  
Three
```

```
20. Condition(?) :)

package javabeginner;

import java.util.Scanner;

public class Conditional
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int num1, num2;
        System.out.print("Enter two numbers: ");
        num1 = input.nextInt();
        num2 = input.nextInt();

        int large = (num1>num2)?num1:num2;
        System.out.println("Large number is: "+large);

    }
}
```

Output:

Enter two numbers: 12 5

Large number is: 12

21. Bitwise and, or, not

```
package javabeginner;
public class Bitwise
{
    public static void main(String[] args)
    {
        int a = 32;
        int b = 12;

        int c;

        c = a & b;
        System.out.println("c = "+c);
        c = a | b;
        System.out.println("c = "+c);
        c = a ^ b;
        System.out.println("c = "+c);
    }
}
```

Output:

```
c = 0
c = 44
c = 44
```

```
22. Binary operator(>>, <<)
package javabeginner;

public class Bitwise
{
    public static void main(String[] args)
    {
        int a = 32;
        int b = 12;

        int c;

        c = a >> 3;
        System.out.println("c = "+c);
        c = a << 3;
        System.out.println("c = "+c);

    }
}

/*
right shift(>>) dara bujay 32 k 2 dara 2 bar vag
left shift(<<) dara bujay 32 k 2 dara 3 bar gun
*/
Output:
c = 4
c = 256
```

```
23. Math Class(Different method of math class)
package javabeginner;
public class MathClass1
{
    public static void main(String[] args)
    {
        int x = 5;
        int y = 2;

        int max = Math.max(x, y);
        System.out.println("Maximum = "+max);

        int min = Math.min(x, y);
        System.out.println("Minimum = "+min);

        int a = -7;
        int abs = Math.abs(a);
        System.out.println("Absolute value = "+abs);

        double power = Math.pow(x, y);
        System.out.println("Power is = "+power);

        int round = Math.round(4.7f);
        System.out.println("Round number is = "+round);

        double pi = Math.PI;
        System.out.println("Value of pi is = "+pi);
    }
}
```

Output:

Maximum = 5

Minimum = 2

Absolute value = 7

Powe is = 25.0

Round number is = 5

Value of pi is = 3.141592653589793

24. For loop

```
package javabeginner;

public class MathClass1
{
    public static void main(String[] args)
    {
        for(int i = 1; i <= 10; i++){
            System.out.println(i+".Bangladesh");
        }
    }
}
```

Output:

```
1.Bangladesh
2.Bangladesh
3.Bangladesh
4.Bangladesh
5.Bangladesh
6.Bangladesh
7.Bangladesh
8.Bangladesh
9.Bangladesh
10.Bangladesh
```

25. While loop

```
package javabeginner;

public class MathClass1
{
    public static void main(String[] args)
    {
        int i = 1;
        while(i<=10){
            System.out.println(i+".Bangladesh");
            i++;
        }
    }
}
```

26. do while loop

```
package javabeginner;
```

```
public class MathClass1
```

```
{
```

```
    public static void main(String[] args)
```

```
{
```

```
        int i = 1;
```

```
        do{
```

```
            System.out.println(i+".Bangladesh");
```

```
            i++;
```

```
        }while(i<=10);
```

```
}
```

```
}
```

27.1 Break and Continue Statement

```
package javabeginner;

public class BreakContinue
{
    public static void main(String[] args)
    {
        int i;
        for(i = 1; i<=100; i++){
            if(i==10)
                break;
            System.out.println(i);
        }
    }
}
```

Output:

```
1
2
3
4
5
6
7
8
9
```

27.2 Break and Continue Statement

```
package javabeginner;
public class BreakContinue
{
    public static void main(String[] args)
    {
        int i;
        for(i = 1; i<=15; i++){
            if(i==10)
                continue;
            System.out.println(i);
        }
    }
}
```

Output:

```
1
2
3
4
5
6
7
8
9
11
12
13
14
15
```

27.3 Break and Continue Statement

```
package javabeginner;
public class BreakContinue
{
    public static void main(String[] args)
    {
        int i;
        for(i = 1; i<=15; i++){
            if(i==10)
                continue;
            if(i>13)
                break;
            System.out.println(i);
        }
    }
}
```

Output:

```
1
2
3
4
5
6
7
8
9
11
12
13
```

28. Print sum of all the number from 1 to 10

```
package javabeginner;
```

```
public class Sum
{
    public static void main(String[] args)
    {
        int sum = 0;
        for (int i = 0; i < 10; i++) {
            sum = sum + i;
        }
        System.out.println("Sum is = "+sum);
    }
}
```

Output: Sum is = 45

29. Print sum of all the number from m to n

```
package javabeginner;

import java.util.Scanner;

public class Sum
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int m, n;
        System.out.print("Enter the initial value: ");
        m = input.nextInt();
        System.out.print("Enter the final value: ");
        n = input.nextInt();
        int sum = 0;

        for (int i = m; i < n; i++) {
            sum = sum + i;
        }
        System.out.println("Sum is = "+sum);
    }
}
```

Output:

```
Enter the initial value: 5
Enter the final value: 10
Sum is = 35
```

30. Print sum of all the even number from m to n

```
package javabeginner;

import java.util.Scanner;

public class Sum
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int m, n;
        System.out.print("Enter the initial value: ");
        m = input.nextInt();
        System.out.print("Enter the final value: ");
        n = input.nextInt();
        int sum = 0;

        for (int i = m; i <= n; i++) {
            if(i%2==0){      //i!=2 dily odd number gulo print hoby.
                sum = sum + i;
                System.out.print(" "+i);
            }
        }
        System.out.println();
        System.out.println("Sum is = "+sum);
    }
}
```

Output:

Enter the initial value: 1

Enter the final value: 10

2 4 6 8 10

Sum is = 30

31. Print sum of $1+2+3+\dots+n$

```
package javabeginner;

import java.util.Scanner;

public class Sum
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int m;
        System.out.print("Enter the final value: ");
        m = input.nextInt();
        int sum = 0;
        for (int i = 1; i <= m; i++) {
            sum = sum + i;
            System.out.print(" "+i);
        }
        System.out.println();
        System.out.println("Sum is = "+sum);
    }
}
```

Output:

```
Enter the final value: 10
```

```
1 2 3 4 5 6 7 8 9 10
```

```
Sum is = 55
```

32. Print sum of 2+4+6+-----+n

```
package javabeginner;
```

```
import java.util.Scanner;
```

```
public class Sum
```

```
{
```

```
    public static void main(String[] args)
```

```
{
```

```
    Scanner input = new Scanner(System.in);
```

```
    int m;
```

```
    System.out.print("Enter the final value: ");
```

```
    m = input.nextInt();
```

```
    int sum = 0;
```

```
    for (int i = 2; i <= m; i = i + 2) {
```

```
        sum = sum + i;
```

```
        System.out.print(" "+i);
```

```
}
```

```
    System.out.println();
```

```
    System.out.println("Sum is = "+sum);
```

```
}
```

```
}
```

Output:

```
Enter the final value: 10
```

```
2 4 6 8 10
```

```
Sum is = 30
```

33. Print sum of $1+3+5+\dots+n$

```
package javabeginner;

import java.util.Scanner;

public class Sum
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int m;
        System.out.print("Enter the final value: ");
        m = input.nextInt();
        int sum = 0;
        for (int i = 1; i <= m; i = i + 2) {
            sum = sum + i;
            System.out.print(" "+i);
        }
        System.out.println();
        System.out.println("Sum is = "+sum);
    }
}
```

Output:

```
Enter the final value: 10
```

```
1 3 5 7 9
```

```
Sum is = 25
```

34. Print sum of $1.5+2.5+3.5+\dots+n$

```
package javabeginner;

import java.util.Scanner;

public class Sum
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        double m, sum = 0;
        System.out.print("Enter the final value: ");
        m = input.nextDouble();

        for (double i = 1.5; i <= m; i++) {
            sum = sum + i;
            System.out.print(" "+i);
        }
        System.out.println();
        System.out.println("Sum is = "+sum);
    }
}
```

Output:

```
Enter the final value: 10
1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5
Sum is = 49.5
```

35. Print sum of $1^2+2^2+3^2+\dots+n^2$

```
package javabeginner;

import java.util.Scanner;

public class Sum
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int m, sum = 0;
        System.out.print("Enter the final value: ");
        m = input.nextInt();

        for (int i = 1; i <= m; i++) {
            sum = sum + i * i;
            System.out.print(i+"x"+i+" ");
        }
        System.out.println();
        System.out.println("Sum is = "+sum);
    }
}
```

Output:

```
Enter the final value: 5
```

```
1x1 2x2 3x3 4x4 5x5
```

```
Sum is = 55
```

36. Print sum of $1*2*3*...*n$

```
package javabeginner;

import java.util.Scanner;

public class Multiple
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int m, result = 1;
        System.out.println("Enter the last value: ");
        m = input.nextInt();

        for (int i = 1; i <= m; i++) {
            result = result * i;
            System.out.print(i+" ");
        }
        System.out.println();
        System.out.println("Result is = "+result);
    }
}
```

Output:

Enter the last value:

5

1 2 3 4 5

Result is = 120

37. Factorial printing

```
package javabeginner;

import java.util.Scanner;

public class Multiple
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int num, fact = 1;
        System.out.print("Enter any positive intergre: ");
        num = input.nextInt();

        for (int i = num; i >= 1; i--)
        {
            fact = fact * i;
        }
        System.out.println();
        System.out.println("Factorial of" + num + " is = " + fact);
    }
}
```

Output:

```
Enter any positive intergre: 4
```

```
Factorial of 4 is = 24
```

38.1 Multiplicaton table

```
package javabeginner;

import java.util.Scanner;

public class Multiple
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);

        int m, result;
        System.out.print("Enter the value: ");
        m = input.nextInt();
        System.out.println("The multiplicaton table of"+m +"is ");

        for (int i = 1; i <= 10; i++) {
            result = m * i;
            System.out.println(+m +" x " +i +" = " +result);
        }
    }
}
```

Output:

Enter the value: 10

The multiplicaton table of10is

10 x 1 = 10

10 x 2 = 20

10 x 3 = 30

10 x 4 = 40

10 x 5 = 50

10 x 6 = 60

10 x 7 = 70

10 x 8 = 80

10 x 9 = 90

10 x 10 = 100

```
38.2 Multiplicaton table*/
package javabeginner;

import java.util.Scanner;

public class Multiple
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);

        int m, result, n;
        System.out.print("Enter the initial value: ");
        m = input.nextInt();
        System.out.print("Enter the final value: ");
        n = input.nextInt();

        for(int j = m; j<=n; j++){
            for (int i = 1; i <= 10; i++) {
                result = j * i;
                System.out.println(+j +" x " +i +" = " +result);
            }
            System.out.println("\n");
        }
    }
}
```

Enter the initial value: 1

Enter the final value: 3

1 x 1 = 1

1 x 2 = 2

1 x 3 = 3

1 x 4 = 4

1 x 5 = 5

1 x 6 = 6

1 x 7 = 7

1 x 8 = 8

1 x 9 = 9

1 x 10 = 10

2 x 1 = 2

2 x 2 = 4

2 x 3 = 6

2 x 4 = 8

2 x 5 = 10

2 x 6 = 12

2 x 7 = 14

2 x 8 = 16

2 x 9 = 18

2 x 10 = 20

3 x 1 = 3

3 x 2 = 6

3 x 3 = 9

3 x 4 = 12

3 x 5 = 15

3 x 6 = 18

3 x 7 = 21

3 x 8 = 24

3 x 9 = 27

3 x 10 = 30

39. Prime or not

```
package javabeginner;
import java.util.Scanner;
public class PrimeNumber
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);

        int num;
        System.out.print("Enter any number: ");
        num = input.nextInt();
        int count = 0;

        for (int i = 2; i < num; i++) {
            if(num%i==0){
                count++;
                break;
            }
        }
        if(count == 0)
            System.out.println("Prime number");
        else
            System.out.println("Not a prime number");
    }
}
```

Output:

Enter any number: 5

Prime number

40. Prime number from m to n

```
package javabeginner;

import java.util.Scanner;

public class PrimeNumber
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);

        int m, n, count = 0, totalPrime = 0;
        System.out.println("Enter initial value: ");
        m = input.nextInt();
        System.out.println("Enter final value: ");
        n = input.nextInt();

        for(int i = m; i<=n; i++)
        {
            for(int j = 2; j<i; j++)
            {
                if(i%j==0)
                {
                    count++;
                    break;
                }
            }
            if(count==0){
                System.out.println(+i);
                totalPrime++;
            }
            count = 0;
        }
        System.out.println("Total prime number = "+totalPrime);
    }
}
```

Output:

Enter initial value:

3

Enter final value:

50

3

5

7

11

13

17

19

23

29

31

37

41

43

47

Total prime number = 1

41. Fibonacci Number Print

```
package javabeginner;

import java.util.Scanner;

public class FibonacceNumbers {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("How many numbers: ");
        int n = input.nextInt();

        int first = 0;
        int second = 1;
        int fibo;

        System.out.print(+first+ " "+second);

        for (int i = 3; i <= n; i++) {
            fibo = first + second;
            first = second;
            second = fibo;
            System.out.print(" "+fibo);
        }
    }
}
```

Output:

```
How many numbers: 7
```

```
0 1 1 2 3 5 8
```

42. Sum of digit Print

```
package javabeginner;

import java.util.Scanner;

public class SumofDigit {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        int n;
        System.out.print("Enter a number: ");
        n = input.nextInt();

        int temp = n;
        int sum = 0, r;

        while(temp != 0)
        {
            r = temp % 10;
            sum = sum + r;
            temp = temp / 10;
        }
        System.out.println("The sum of digit is = "+sum);
    }
}
```

Output:

Enter a number: 244

The sum of digit is = 10

43. Reverse a number and print it

```
package javabeginner;

import java.util.Scanner;

public class Reverse {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        int n;
        System.out.print("Enter a number: ");
        n = input.nextInt();

        int temp = n;
        int sum = 0, r;

        while(temp != 0)
        {
            r = temp % 10;
            sum = sum * 10 + r;
            temp = temp / 10;
        }
        System.out.println("The reverse of the number is = "+sum);
    }
}
```

Output:

Enter a number: 123

The reverse of the number is = 321

```
44. Palindrome number print

package javabeginner;
import java.util.Scanner;
public class Palindrome {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        int n;
        System.out.print("Enter a number: ");
        n = input.nextInt();

        int temp = n;
        int sum = 0, r;

        while(temp != 0)
        {
            r = temp % 10;
            sum = sum * 10 + r;
            temp = temp / 10;
        }
        if(n==sum)
            System.out.println("This is a palindrome number");
        else
            System.out.println("This is not a palindrome number");
    }
}
```

Output:

Enter a number: 121

This is a palindrome number

```
45. Armstrong number print
package javabeginner;
import java.util.Scanner;
public class Armstrong {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        int n;
        System.out.print("Enter a number: ");
        n = input.nextInt();

        int temp = n;
        int sum = 0, r;

        while(temp != 0)
        {
            r = temp % 10;
            sum = sum + r * r * r;
            temp = temp / 10;
        }
        if(n==sum)
            System.out.println("This is a armstrong number");
        else
            System.out.println("This is not a armstrong number");
    }
}
```

Output:

```
Enter a number: 153
This is a armstrong number
```

46.1 Pattern print

```
package javabeginner;

import java.util.Scanner;

public class Pattern {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        int n;
        System.out.print("Enter row number: ");
        n = input.nextInt();

        for (int row = 1; row <= n; row++) {
            for(int col = 1; col <= row; col++){
                System.out.print(" " +col);
            }
            System.out.println("");
        }
    }
}
```

Output:

```
Enter row number: 4
```

```
1
1 2
1 2 3
1 2 3 4
```

46.2 Pattern print

```
package javabeginner;

import java.util.Scanner;

public class Pattern {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        int n;
        System.out.print("Enter row number: ");
        n = input.nextInt();

        for (int row = n; row >= 1; row--) {
            for(int col = 1; col <= row; col++){
                System.out.print(" " +col);
            }
            System.out.println("");
        }
    }
}
```

Output:

```
Enter row number: 4
1 2 3 4
1 2 3
1 2
1
```

47. Array sum and average

```
package javabeginner;
```

```
public class JavaBeginner {
```

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

```
        int[] number = new int[5];
```

```
        number[0] = 10;
```

```
        number[1] = 20;
```

```
        number[2] = 30;
```

```
        number[3] = 40;
```

```
        number[4] = 50;
```

```
        int sum = number[0] + number[1] + number[2] + number[3] + number[4];
```

```
        double avg = sum / number.length;
```

```
        int length = number.length;
```

```
        System.out.println("Sum is = " + sum);
```

```
        System.out.println("Average is = " + avg);
```

```
        System.out.println("Length is = " + length);
```

```
}
```

```
}
```

Output:

Sum is = 150

Average is = 30.0

Length is = 5

48.1 Array sum and average from the user

```
package javabeginner;

import java.util.Scanner;
public class JavaBeginner {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        double[] number = new double[5];
        double sum = 0;

        System.out.println("Please enter 5 number = ");
        for (int i = 0; i < 5; i++) {
            number[i] = input.nextDouble();
        }

        for (int i = 0; i < 5; i++) {
            sum = sum + number[i];
        }

        double length = number.length;
        double avg = sum / number.length;

        System.out.println("Sum is = " + sum);
        System.out.println("Average is = " + avg);
        System.out.println("Length is = " + length);

    }
}
```

Output:

Please enter 5 number =

12 23 34 45 56

Sum is = 170.0

Average is = 34.0

Length is = 5.0

48.2 Array sum and average from the user

```
package javabeginner;
import java.util.Scanner;
public class JavaBeginner {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("How many numbers: ");
        int n;
        n = input.nextInt();
        double[] number = new double[n];
        double sum = 0;

        System.out.println("Please enter "+n+" number = ");
        for (int i = 0; i < n; i++) {
            number[i] = input.nextDouble();
        }
        for (int i = 0; i < n; i++) {
            sum = sum + number[i];
        }

        double length = number.length;
        double avg = sum / number.length;

        System.out.println("Sum is = " + sum);
        System.out.println("Average is = " + avg);
        System.out.println("Length is = " + length);
    }
}
```

Output:

How many numbers: 5

Please enter 5 number =

1 2 3 4 5

Sum is = 15.0

Average is = 3.0

Length is = 5.0

49. Array-Finding maximum and minimum from the user
package javabeginner;

```
import java.util.Scanner;

public class JavaBeginner {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("How many numbers: ");
        int n;
        n = input.nextInt();
        double[] number = new double[n];
        double sum = 0;

        System.out.println("Please enter " + n + " number = ");
        for (int i = 0; i < n; i++) {
            number[i] = input.nextDouble();
        }

        for (int i = 0; i < n; i++) {
            sum = sum + number[i];
        }

        double length = number.length;
        double avg = sum / number.length;
```

```
System.out.println("Sum is = " + sum);
System.out.println("Average is = " + avg);
System.out.println("Length is = " + length);

double max = number[0];
double min = number[0];
for (int i = 1; i < number.length; i++) {
    if (max < number[i]) {
        max = number[i];
    }

    if (min > number[i]) {
        min = number[i];
    }
}

System.out.println("Maximum is = " + max);
System.out.println("Minimum is = " + min);

}
```

Output:

```
How many numbers: 5
Please enter 5 number =
12 21 23 11 45
Sum is = 112.0
Average is = 22.4
Length is = 5.0
Maximum is = 45.0
Minimum is = 11.0
```

50. Printing string

```
public class Main {  
  
    public static void main(String[] args) {  
  
        String[] name = {"Kibria", "jamal", "Ismail", "dasklf", "dlf"};  
        for (int i = 0; i < 5; i++) {  
            System.out.println(name[i]);  
        }  
    }  
}
```

Output:

```
kibria  
jamal  
Ismail  
dasklf  
dlf
```

51. Print string using for each loop

```
public class Main {  
  
    public static void main(String[] args) {  
  
        String[] name = {"Kibria", "jamal", "Ismail", "dasklf", "dlf"};  
        for (String x : name) {  
            System.out.println(x);  
        }  
    }  
}  
  
Output:  
Output:  
kibria  
jamal  
Ismail  
dasklf  
dlf
```

52. Print a simple matrix

```
package matrix;

public class Matrix {

    public static void main(String[] args) {
        int[][] number = new int[2][2];

        number[0][0] = 10;
        number[0][1] = 20;
        number[1][0] = 30;
        number[1][1] = 40;

        /*
        System.out.println(number[0][0]);
        System.out.println(number[0][1]);
        System.out.println(number[1][0]);
        System.out.println(number[1][1]);
        */

        for (int row = 0; row < 2; row++) {
            for (int col = 0; col < 2; col++) {
                System.out.print(" " + number[row][col]);
            }
            System.out.println("");
        }
    }
}
```

Output:

```
10 20
30 40
```

53. Printing a matrix and its sum

```
package matrix;

import java.util.Scanner;

public class Matrix {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        int[][] A = new int[2][2];
        int[][] B = new int[2][2];

        /*TakeInput*/
        System.out.println("Enter element of A matrix: ");
        for (int row = 0; row < 2; row++) {
            for (int col = 0; col < 2; col++) {
                System.out.printf("A[%d][%d] = ", row, col);
                A[row][col] = input.nextInt();
            }
        }
        System.out.println("Enter element of B matrix: ");
        for (int row = 0; row < 2; row++) {
            for (int col = 0; col < 2; col++) {
                System.out.printf("B[%d][%d] = ", row, col);
                B[row][col] = input.nextInt();
            }
        }

        /*PrintingOutput*/
        System.out.println("");
        System.out.print("A = ");
        for (int row = 0; row < 2; row++) {
            for (int col = 0; col < 2; col++) {
                System.out.print(" \t" + A[row][col]);
            }
            System.out.println("");
        }
        System.out.println("");
        System.out.print("B = ");
        for (int row = 0; row < 2; row++) {
            for (int col = 0; col < 2; col++) {
                System.out.print(" \t" + B[row][col]);
            }
            System.out.println("");
        }
    }
}
```

```

        System.out.println("Sum of the two matrix is : ");
        for (int row = 0; row < 2; row++) {
            for (int col = 0; col < 2; col++) {
                System.out.print(" \t" + (A[row][col] + B[row][col]));
            }
            System.out.println("");
        }

        /*
        We can also print the sum by this method:
        int[][] C = new int[2][2];
        C[row][col] = A[row][col]+B[row][col];
        System.out.println(" \t"+C[row][col]);
        */
    }
}

```

Output:

Enter element of A matrix:

```

A[0][0] = 1
A[0][1] = 2
A[1][0] = 3
A[1][1] = 4

```

Enter element of B matrix:

```

B[0][0] = 5
B[0][1] = 6
B[1][0] = 7
B[1][1] = 8

```

```

A =   1   2
      3   4

```

```

B =   5   6
      7   8

```

Sum of the two matrix is :

```

6   8
10  12

```

54. Printing the sum of diagonal, upper and lower triangle element.

```
package matrix;
import java.util.Scanner;
public class Matrix {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        int[][] A = new int[3][3];

        /*TakeInput*/
        System.out.println("Enter element of the matrix: ");
        for (int row = 0; row < 3; row++) {
            for (int col = 0; col < 3; col++) {
                System.out.printf("A[%d][%d] = ", row, col);
                A[row][col] = input.nextInt();
            }
        }
        /*PrintingOutput*/
        System.out.println("");
        System.out.print("A = ");
        for (int row = 0; row < 3; row++) {
            for (int col = 0; col < 3; col++) {
                System.out.print(" \t" + A[row][col]);
            }
            System.out.println("");
        }

        System.out.println("");
        int sum = 0, utriangle = 0, ltriangle = 0;
        for (int row = 0; row < 3; row++) {
            for (int col = 0; col < 3; col++) {
                if (row == col) {
                    sum = sum + A[row][col];
                }
                if (row < col) {
                    utriangle = utriangle + A[row][col];
                }
                if (row > col) {
                    ltriangle = ltriangle + A[row][col];
                }
            }
        }
        System.out.println("Sum of Diagonal Elements = " + sum);
        System.out.println("Sum of UpperTri Elements = " + utriangle);
        System.out.println("Sum of LowerTri Elements = " + ltriangle);
    }
}
```

Output:

Enter element of the matrix:

A[0][0] = 1

A[0][1] = 2

A[0][2] = 3

A[1][0] = 4

A[1][1] = 5

A[1][2] = 6

A[2][0] = 7

A[2][1] = 8

A[2][2] = 9

A = 1 2 3
4 5 6
7 8 9

Sum of Diagonal Elements = 15

Sum of UpperTri Elements = 11

Sum of LowerTri Elements = 19

55.1 Printing Array List

```
package matrix;

public class Array {

    public static void main(String[] args) {
        int[][] number = new int[4][5];
        int k = 0;

        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 5; j++) {
                number[i][j] = k;
                k++;
            }
        }
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 5; j++) {
                System.out.print(+number[i][j] + " ");
            }
            System.out.println("");
        }
    }
}
```

Output:

```
0 1 2 3 4
5 6 7 8 9
10 11 12 13 14
15 16 17 18 19
```

55.2 Printing Array List

```
package matrix;

public class Array {

    public static void main(String[] args) {
        int[][] number = new int[4][];
        int k = 0;

        /*Orthat zero number row a column thakby ak ta*/
        number[0] = new int[1];
        number[1] = new int[2];
        number[2] = new int[3];
        number[3] = new int[4];

        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < i+1; j++) {
                number[i][j] = k;
                k++;
            }
        }

        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < i+1; j++) {
                System.out.print(+number[i][j] + " ");
            }
            System.out.println("");
        }
    }
}
```

Output:

0

1 2

3 4 5

6 7 8 9

56. Array Sorting

```
package matrix;

import java.util.Arrays;
import java.util.Scanner;

public class ArraySorting {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        System.out.print("How many numbers? ");
        int n = input.nextInt();
        System.out.printf("Enter %d numbers: ", n);
        int[] number = new int[n];
        for (int i = 0; i < n; i++) {
            number[i] = input.nextInt();
        }

        // int[] number = {10, -3, 18, 5, 25};
        Arrays.sort(number);

        System.out.print("Ascending order: ");
        for (int i = 0; i < n; i++) {
            System.out.print(" " + number[i]);
        }
        System.out.println("");
    }
}
```

```
System.out.print("Descending order: ");
for (int i = n - 1; i >= 0; i--) {
    System.out.print(" " + number[i]);
}
System.out.println("");
}
```

Output:

```
How many numbers? 5
Enter 5 numbers: 12 23 32 -9 43
Ascending order: -9 12 23 32 43
Descending order: 43 32 23 12 -9
```

57. Printing ArrayList Using Normal/foreach/Iterator method

```
package javaapplication;
import java.util.ArrayList;
import java.util.Iterator;

public class JavaApplication {

    public static void main(String[] args) {
        ArrayList<Integer> number = new ArrayList<Integer>();
        System.out.println("Size = " + number.size());

        number.add(10);
        number.add(20);
        number.add(30);
        number.add(40);
        number.add(4, 50);

        /*
            Printing the output using for each Loop:
            for(int x : number){
                System.out.print(+x+ " ");
            }
        */

        /*
            Printing the output using normal way
            System.out.println(number);
        */

        System.out.print("ArrayList Contain: ");
        Iterator itr = number.iterator();
        while (itr.hasNext()) {
            System.out.print(itr.next() + " ");
        }
        System.out.println("");
        System.out.println("Size = " + number.size());

        number.remove(2);
        System.out.println("After removing one element: " + number);

        number.removeAll(number);
        System.out.println("After removing all elements: " + number);
    }
}
```


58. Printing ArrayList Using Normal/foreach/Iterator method.

```
package javaapplication;

import java.util.ArrayList;

public class JavaApplication {

    public static void main(String[] args) {
        ArrayList<Integer> number = new ArrayList<Integer>();
        System.out.println("Size = " + number.size());

        number.add(10);
        number.add(20);
        number.add(30);
        number.add(40);
        number.add(4, 50);

        System.out.print("ArrayList Contain: "+number);

        System.out.println("");
        System.out.println("Size = " + number.size());

        /* number.clear();
           System.out.println("After clearing arrayList contain: "+number); */

        boolean check = number.isEmpty();
        System.out.println("ArrayList Empty: " + check);
        /*arraylist empty true dekacci cuz amra age array list ta clear kory
        niyechi number.clear(); er maddomy */

        boolean b = number.contains(30);
        System.out.println("This element is in the list: " + b);

        int pos = number.indexOf(40);
        System.out.println("The index is : " + pos);

        /* Ay method er maddomy mulotw index er value taky replace kora
        jay. Orthat 3 number index a je 40 asy setaky ami 35 dara replace korbo
        */
        number.set(3, 35);
        System.out.println("After repalce index three: "+number);

        int x = number.get(0);
        System.out.println("Index zero is : "+x);
    }
}
```



```

59. ArrayList
package javaapplication;

import java.util.ArrayList;

public class Array {

    public static void main(String[] args) {
        ArrayList<Integer> number1 = new ArrayList<>();
        ArrayList<Integer> number2 = new ArrayList<>();
        ArrayList<Integer> number3 = new ArrayList<>();

        number1.add(10);
        number1.add(20);
        number1.add(30);
        number1.add(40);
        System.out.println("Number1 = "+number1);

        number2.add(50);
        number2.add(60);
        number2.add(70);
        number2.add(80);
        System.out.println("Number2 = "+number2);

        number3.addAll(number1);
        System.out.println("Number3 = "+number3);

        boolean b = number1.equals(number2);
        System.out.println("Number1 is equal number2 = "+b);

        boolean c = number1.equals(number3);
        System.out.println("Number1 is equal number3 = "+c);
    }
}

```

Output:

```

Number1 = [10, 20, 30, 40]
Number2 = [50, 60, 70, 80]
Number3 = [10, 20, 30, 40]
Number1 is equal number2 = false
Number1 is equal number3 = true

```

60. Sorting ArrayList

```
package javaapplication;

import java.util.ArrayList;
import java.util.Collections;

public class Array {

    public static void main(String[] args) {
        ArrayList<Integer> number = new ArrayList<>();

        number.add(10);
        number.add(20);
        number.add(13);
        number.add(17);
        number.add(-2);
        number.add(12);

        System.out.println("Before Sorting = "+number);

        Collections.sort(number);
        System.out.println("After sorting ascending order = "+number);

        Collections.sort(number, Collections.reverseOrder());
        System.out.println("After sorting descending order = "+number);
    }
}
```

Output:

```
Before Sorting = [10, 20, 13, 17, -2, 12]
After sorting ascending order = [-2, 10, 12, 13, 17, 20]
After sorting descending order = [20, 17, 13, 12, 10, -2]
```

61.1 String print

```

package javabeginners;

public class JavaBeginners {

    public static void main(String[] args) {

        String s1 = "Golam kibria";
        String s2 = new String("Golam kibria");

        char[] ch = {'e', 'z', 'a', 'z'};
        System.out.println(ch);

        System.out.println("s1 = " + s1);
        System.out.println("s2 = " + s2);

        int len = s1.length();
        System.out.println("Length is = " + len);

        /*equalsIgnoreCase(s2)*/
        if (s1.equals(s2)) {
            System.out.println("Equals");
        } else {
            System.out.println("Not equals");
        }

        boolean b = s1.contains(s2);
        System.out.println("s1 contains s2 = " + b);

        boolean c = s1.contains("kibria");
        System.out.println("s1 contains kibria = " + c);

        boolean d = s1.isEmpty();
        System.out.println("s1 is empty = " + d);
    }
}

```

Output:

```

ezaz
s1 = Golam kibria
s2 = Golam kibria
Length is = 12
Equals
s1 contains s2 = true
s1 contains kibria = true
s1 is empty = false

```

61.2 String print

```
package javabeginners;

public class JavaBeginners {

    public static void main(String[] args) {

        String firstname = "Golam";
        String lastname = " kibria";

        String fullname = firstname + lastname;
        System.out.println("Fullname is = " + fullname);

        /*String fullname = firstname.concat(lastname);
        System.out.println("Fullname is = "+fullname);*/
        String upper = fullname.toUpperCase();
        System.out.println("Uppercase name is = " + upper);

        String lower = fullname.toLowerCase();
        System.out.println("Lowercase name is = " + lower);

        boolean b = firstname.startsWith("G");
        System.out.println("b = " + b);

        boolean c = lastname.endsWith("ia");
        System.out.println("c = " + c);

        String[] name = {"Golam", " Kibria"};
        for (String x : name) {
            System.out.print(x);
        }
        System.out.println("");
        /*printing using for Loop*/
        for (int i = 0; i < 2; i++) {
            System.out.print(name[i]);
        }
        System.out.println("");
    }
}
```

Output:

```
Fullname is = Golam kibria
Uppercase name is = GOLAM KIBRIA
Lowercase name is = golam kibria
b = true
c = true
Golam Kibria
Golam Kibria
```

61.3 String print

```
package string;

public class String {

    public static void main(String[] args) {

        String country = " My country name is Bangladesh ";
        System.out.println(country);

        /* Space gula k remove kory diby */
        String s = country.trim();
        System.out.println(s);

        char ch = country.charAt(0);
        System.out.println(ch);

        int value = country.codePointAt(0);
        System.out.println(value);

        int pos = country.indexOf("y");
        System.out.println(pos);

        pos = country.lastIndexOf("n");
        System.out.println(pos);

    }
}
```

Output:

My country name is Bangladesh

My country name is Bangladesh

32

2

22

61.4 String part

```
package stringcode;
```

```
public class StringCode {
```

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

```
        String s1 = " This is my country ";
```

```
        System.out.println(s1);
```

```
        String s2 = s1.replace('i', 'j');
```

```
        System.out.println(s2);
```

```
/* Space er upor vitti kory string taky vag korychy */
```

```
        String[] s3 = s1.split(" ");
```

```
        for (String x : s3) {
```

```
            System.out.println(x);
```

```
        }
```

```
}
```

Output:

```
This is my country
```

```
Thjs js my country
```

This

is

my

country

62. String Buffer

```
package stringcode;
```

```
public class StringCode {
```

```
    public static void main(String[] args) {  
  
        /*String s = "Golam kibria"; */  
        StringBuffer s = new StringBuffer("Golam kibria");  
        /*StringBuffer s = new StringBuffer(s); */  
        System.out.println(s);  
  
        s.append(" Ezaz");  
        s.append(" 25");  
        System.out.println(s);  
  
        s.setLength(5);  
        System.out.println(s);  
  
        s.delete(0, 3);  
        System.out.println(s);  
    }  
}
```

Output:

```
Golam kibria  
Golam kibria Ezaz 25  
Golam  
am
```

63. Checking palindrome or not

```
package stringcode;

public class StringCode {

    public static void main(String[] args) {

        String s1 = "madam";

        StringBuffer sb = new StringBuffer(s1);
        String s2 = sb.reverse().toString();

        if(s1.equals(s2)){
            System.out.println("Palindrome");
        }else{
            System.out.println("Not a palindrome");
        }
    }
}
```

Output:

Palindrome

64.1 `StringBuilder`

```
package stringjava;

public class StringJava {

    public static void main(String[] args) {

        StringBuilder str = new StringBuilder("Golam");
        System.out.println("str = " + str);

        str.append(" Kibria ");
        str.append(20);
        str.append(21);

        System.out.println("str = " + str);

        str.delete(2, 4);
        System.out.println("str = " + str);

        str.reverse();
        System.out.println("str = " + str);

    }
}
```

Output:

```
str = Golam
str = Golam Kibria 2021
str = Gom Kibria 2021
str = 1202 airbiK moG
```

64.2 *StringBuilder*

```
package stringjava;

public class StringJava {

    public static void main(String[] args) {

        //primitive -> object
        int x = 20;
        Integer y = Integer.valueOf(x);
        System.out.println("y = "+y);
        Integer z = x; //Integer.valueOf(x) autoboxing
        System.out.println("y = "+z);

        //object -> primitive
        Double d = new Double(10.25);
        System.out.println("d = "+d);
        Double e = d;
        System.out.println("e = "+e);

        //integer -> string
        int i = 100;
        String s = Integer.toString(i);
        /*double diyeo kora jay tokon Integer er poriborty Double hoby*/
        System.out.println("s = "+s);

        boolean b = true;
        String str = Boolean.toString(b);
        System.out.println("str = "+str);

        //string -> integer
        String st = "12";
        int j = Integer.parseInt(st);
        System.out.println("j = "+j);
        /*double diyeo kora jay tokon Integer er poriborty Double hoby*/
    }
}
```


65. NumberConversion(Decimal, Binary)

```
package stringjava;

public class StringJava {

    public static void main(String[] args) {
        //binary to decimal, octal & hexa
        String binary = "1010";
        Integer decimal1 = Integer.parseInt(binary, 2);
        System.out.println("Decimal value = " + decimal1);

        String octal = "543";
        Integer decimal2 = Integer.parseInt(octal, 8);
        System.out.println("Decimal value = " + decimal2);

        String hexadecimal = "A";
        Integer decimal3 = Integer.parseInt(hexadecimal, 16);
        System.out.println("Decimal value = " + decimal3);

        //decimal to binary, octal & hexa
        System.out.println("");
        int decimal = 15;
        String binary1 = Integer.toBinaryString(decimal);
        System.out.println("Binary value is = " + binary1);

        int decimall = 15;
        String octall = Integer.toOctalString(decimall);
        System.out.println("Octal value is = " + octall);

        int dec1 = 15;
        String hex = Integer.toHexString(dec1);
        System.out.println("Hexadecimal value is = " + hex);
    }
}
```


66. Printing Date

```
package datetime;

import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;

public class DateTime {

    public static void main(String[] args) {

        Date d = new Date();
        //System.out.println(d);

        DateFormat df = new SimpleDateFormat("dd/MM/YYYY");
        String currentdate = df.format(d);
        System.out.println("Current Date: "+currentdate);
    }
}
```

Output:

Current Date: 04/10/2021

67. TimePrinting

```
package datetime;

import java.time.LocalTime;
import java.time.format.DateTimeFormatter;

public class Time {

    public static void main(String[] args) {

        LocalTime t = LocalTime.now();
        //System.out.println(t);

        DateTimeFormatter dt =
DateTimeFormatter.ofPattern("hh:mm:ss");
        String time = t.format(dt);
        System.out.println("Current time = "+time);
    }
}
```

Output:

```
Current time = 04:26:10
```

68. RandomNumber

```
package javaapplication1;

import java.util.Random;

public class JavaApplication1 {

    public static void main(String[] args) {

        Random r = new Random();
        int randomnumber = r.nextInt(10);
        //int randomnumber = r.nextInt(10) + 1 diyeo korty pari.
        //tokon r zero show korby na
        System.out.println("Random number: " + randomnumber);

        /*
            int rd = (int) (Math.random() * 10);
            System.out.println("Random number: "+rd);
        */
    }
}
```

Output:

```
Random number: 3
/*Ak ak somoy ak ak number asby*/
```

69. Class creation

Akta package er under a dui ta class thakby.

Class-1 er code:

```
package oop;
public class Teacher {
    String name, gender;
    int age;
}
```

Class-2 er code:

```
package oop;

public class Test {

    public static void main(String[] args) {
        /*Teacher class er akta object create korychi tarpor sei
        object taky akany declare korychi*/
        Teacher teacher1 = new Teacher();

        teacher1.name = "Golam kibria";
        teacher1.gender = "Male";
        teacher1.age = 20;

        System.out.println("Name: " + teacher1.name);
        System.out.println("Name: " + teacher1.gender);
        System.out.println("Name: " + teacher1.age);
    }
}
```

```
/*Chailly teacher2 er information o print kora jaby. Sekkhetry sudu  
teacher1 er poriborty teacher2 likhy dilei hoby*/
```

Output:

Name: Golam kibria

Name: Male

Name: 20

70. Introducing method

Ager motoi duita class thakby.

Class-1 er code:

```
package oop;
```

```
public class Teacher {
```

```
    String name, gender;
```

```
    int age;
```

```
    void information(){
```

```
        System.out.println("Name: "+name);
```

```
        System.out.println("Name: "+gender);
```

```
        System.out.println("Name: "+age);
```

```
}
```

```
}
```

Class-2 er code:

```
package oop;
```

```
public class Test {
```

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

```
        Teacher teacher1 = new Teacher();
```

```
        teacher1.name = "Golam kibria";
```

```
        teacher1.gender = "Male";
```

```
teacher1.age = 20;  
teacher1.information();  
}  
}
```

Output:

Name: Golam kibria

Name: Male

Name: 20

71. Parametrized method

Class-1 er code:

```
package oop;

public class Teacher {

    String name, gender;
    int age;

    void setinformation(String s, String g, int a){
        name = s;
        gender = g;
        age = a;
    }

    void information(){
        System.out.println("Name: "+name);
        System.out.println("Name: "+gender);
        System.out.println("Name: "+age);
    }
}
```

Class-2 er code:

```
package oop;

public class Test {

    public static void main(String[] args) {
        Teacher teacher1 = new Teacher();
```

```
teacher1.setinformation("Golam kibria", "Male", 20);
teacher1.information();

Teacher teacher2 = new Teacher();
teacher2.setinformation("Mottaki Billah", "Male", 21);
teacher2.information();

}

}
```

Output:

```
Name: Golam kibria
Name: Male
Name: 20
Name: Mottaki Billah
Name: Male
Name: 21
```

72. Constructor

Class-1 er code:

```
package nightcode;
```

```
public class Teacher {
```

```
    String name, gender;
```

```
    int age;
```

```
/*Teacher holo akta constructor. Er nam obboisoi class er saty mil thakty hoby*/
```

```
/*Ata holo parameterized constructor*/
```

```
Teacher(String s, String g, int a) {
```

```
    name = s;
```

```
    gender = g;
```

```
    age = a;
```

```
}
```

```
void displayOutput() {
```

```
    System.out.println("Name: " + name);
```

```
    System.out.println("Gender: " + gender);
```

```
    System.out.println("Age: " + age);
```

```
    System.out.println("");
```

```
}
```

```
}
```

Class-2 er code:

```
package nightcode;

public class Test {

    public static void main(String[] args) {

        /*new operator er sahajjy akany object toiri hocco abong saty
        saty value set hoye jaccy*/
        Teacher t1 = new Teacher("Golam kibria", "Male", 20);
        t1.displayOutput();

        Teacher t2 = new Teacher("Sajib", "Male", 21);
        t2.displayOutput();
    }
}

/*
Constructor holo akta special type of method jeta object k
initialize korty babohar hoy. Ataky call korty hoy na. Ata
automatically call hoby. Atar kono return type o thakby na.

Kono parameter na thakly string er return type hoby null.
Abong integer er jonno hoby zero.
*/
```

Output:

Name: Golam kibria

Gender: Male

Age: 20

Name: Sajib

Gender: Male

Age: 21

***Construct er maddomy kono akti onject k initialize korteci & method er maddomy object er behaviour gulo express korteci.

73. Constructor-2

Class-1 er code:

```
package nightcode;

public class Teacher {

    String name, gender;
    int age;

    Teacher() {
        System.out.println("No value");
    }

    Teacher(String s, String g, int a) {
        name = s;
        gender = g;
        age = a;
    }

    void displayOutput() {
        System.out.println("Name: " + name);
        System.out.println("Gender: " + gender);
        System.out.println("Age: " + age);
        System.out.println("");
    }
}
```

Class-2 er code:

```
package nightcode;

public class Test {

    public static void main(String[] args) {

        Teacher t1 = new Teacher("Golam kibria", "Male", 20);
        t1.displayOutput();

        Teacher t2 = new Teacher("Sajib", "Male", 21);
        t2.displayOutput();

        Teacher t3 = new Teacher();
        t3.displayOutput();
    }
}
```

Name: Golam kibria

Gender: Male

Age: 20

Name: Sajib

Gender: Male

Age: 21

No value

Name: null

Gender: null

Age: 0

74. Overloading constructor

Class-1 er code:

```
package nightcode;

public class Teacher {

    String name, gender;
    int age;

    Teacher() {
        System.out.println("No value");
    }

    Teacher(String s, String g) {
        name = s;
        gender = g;
    }

    Teacher(String s, String g, int a) {
        name = s;
        gender = g;
        age = a;
    }

    void displayOutput() {
        System.out.println("Name: " + name);
        System.out.println("Gender: " + gender);
        System.out.println("Age: " + age);
        System.out.println("");
    }
}
```

```
package nightcode;

public class Test {

    public static void main(String[] args) {

        Teacher t1 = new Teacher();

        Teacher t2 = new Teacher("Sajib", "Male");
        t2.displayOutput();

        Teacher t3 = new Teacher("Golam kibria", "Male", 20);
        t3.displayOutput();
    }
}
```

/*
Overloading constructor: Onk gulo contructor declare
korty parbo but tader parameter onnosoi alada hoty hoby.

*/

Output:

No value

Name: Sajib

Gender: Male

Age: 0

Name: Golam kibria

Gender: Male

Age: 20

75. Returning value

Class-1 er code:

```
package nightcode;

public class ReturnType {

    int square(int value) {
        return value * value;
    }
}
```

Class-2 er code:

```
package nightcode;

public class Return {
```

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

        ReturnType a = new ReturnType();
        int result = a.square(5);
        System.out.println("Result is: " + result);
    }
}
```

Output: Result is: 25

```
system.out.println("Result is: "+a.square(5))
return holo keyword & parameter hisaby 5 patacci.
```

76. Static keyword

Class-1 er code:

```
package nightcode;

public class Student {

    String name;
    int id;
    static String uname = "DIU";

    Student(String n, int a) {
        name = n;
        id = a;
    }

    void displayOutput() {
        System.out.println("Name: " + name);
        System.out.println("ID: " + id);
        System.out.println("University name: " + uname);
        System.out.println("");
    }
}
```

Class-2 er code:

```
package nightcode;

public class Test {

    public static void main(String[] args) {

        Student s1 = new Student("Kibria", 10);
        Student s2 = new Student("Munna", 20);
        s1.displayOutput();
        s2.displayOutput();
    }
}

/*
static keyword memory te alada jayga niby na.
cuz static object er saty somporkito na. Ata class
saty somporkito.
*/
```

Output:

Name: Kibria

ID: 10

University name: DIU

Name: Munna

ID: 20

University name: DIU

77.1 Static variable-1

Class-1 er code:

```
package javacode;
```

```
public class Dhaka {
```

```
    String uname = "DIU";
```

```
}
```

Class-2 er code:

```
package javacode;
```

```
public class Test {
```

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

```
        Dhaka ob = new Dhaka();
```

```
        System.out.println("Name: " + ob.uname);
```

```
}
```

```
}
```

Output:

Name: DIU

77.2 Static variable-2

Class-1 er code:

```
package javacode;
```

```
public class Dhaka {
```

```
    static String uname = "DIU";
```

```
}
```

Class-2 er code:

```
package javacode;
```

```
public class Test {
```

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

```
        System.out.println("Name: " + Dhaka.uname);
```

```
}
```

```
}
```

```
/*
```

```
Akhetry static variable subida holo alada baby r kono  
object declare kora lagcy na.
```

```
*/
```

Output:

```
Name: DIU
```

78. Static variable-3

Class-1 er code:

```
package javacode;

public class Dhaka {

    static int count = 0;

    Dhaka() {
        count++;
    }

    void total() {
        System.out.println("Total student:" + count);
    }
}
```

Class-2 er code:

```
package javacode;

public class Test {

    public static void main(String[] args) {

        Dhaka s1 = new Dhaka();
        s1.total();

        Dhaka s2 = new Dhaka();
        s2.total();
    }
}
```

```
Dhaka s3 = new Dhaka();  
s3.total();  
}  
}
```

Output:

Total student:1

Total student:2

Total student:3

79. Static method

Class-1 er code:

```
package javacode;

public class Dhaka {

    void display1() {
        System.out.println("I am non static");
    }

    static void display2() {
        System.out.println("I am static method");
    }
}
```

Class-2 er code:

```
package javacode;
```

```
public class Test {

    public static void main(String[] args) {

        Dhaka ob = new Dhaka();
        ob.display1();

        Dhaka.display2();
    }
}
```

Output:

I am non static

I am static method

/*

Ay main method holo static member er sobcheye common example.
Main method static hisaby declare kora hoy ay jonno j, er jonno
alada vaby kono object toiri korty hoy na.

*/

80.1 Static block

Class-1 er code:

```
package javacode;

public class Dhaka {

    static int id;
    static String name;

    static {
        id = 10;
        name = "Golam Kibria";
    }

    static void display() {
        System.out.println("ID: " + id);
        System.out.println("Name: " + name);
    }

    /*
        public static void main(String[] args) {

            Dhaka.display();
        }
    Aybabu print korly tokon r class-2 er dorkar hoby na.
    */
}
```

Class-2 er code:

```
package javacode;

public class Test {

    public static void main(String[] args) {

        Dhaka.display();
    }
}
```

Output:

ID: 10

Name: Golam Kibria

80.2 Static block-2

```
package javacode;

public class Dhaka {

    static {
        System.out.println("Static block");
    }

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

Output:

```
Static block
Main method
```

81. Types of variable(Static, Instance, Local)

```
package javacode;
```

```
public class Dhaka {
```

```
    String name;
```

```
    int id;
```

```
    static String uname;
```

```
Dhaka(String n, int i){
```

```
    name = n;
```

```
    id = i;
```

```
}
```

```
void display(){
```

```
    System.out.println("Name: "+name);
```

```
    System.out.println("ID: "+id);
```

```
    System.out.println("University name: "+uname);
```

```
}
```

```
}
```

```
/*
String n, int i holo local variable.
->A variable that is declare inside the method is called
local variable.
->Local variable is declare inside method, constructor or in a
block.
```

```
String name;
int id;
static String uname;
Aygulo holo instance variable.
->A variable that is declare inside the class but
outside any method that is called instance variable.
->It is not declare as static.
```

```
Static or class variable: Ay variable chenar sohoj
upay holo jodi kono varable er
agy static likha thaky taholy seta static variable.
It's can't be declare as local variable.
```

```
*/
```

82. Display volume

Class-1 er code:

```
package javacode;

public class Dhaka {

    double height, width, depth;

    Dhaka(double h, double w, double d) {
        height = h;
        width = w;
        depth = d;
    }

    void displayVolume() {
        double vol = height * width * depth;
        System.out.println("Volume is: " + vol);
    }
}
```

Class-2 er code:

```
package javacode;

public class Test {

    public static void main(String[] args) {

        Dhaka box1 = new Dhaka(10, 20, 30);
        Dhaka box2 = new Dhaka(40, 50, 60);

        box1.displayVolume();
        box2.displayVolume();
    }
}
```

Output:

```
Volume is: 6000.0
Volume is: 120000.0
```

83. Instance variable hiding

Class-1 er code:

```
package javacode;

public class Dhaka {

    double height, width, depth;

    Dhaka(double height, double width, double depth) {
        this.height = height;
        this.width = width;
        this.depth = depth;
    }

    void displayVolume() {
        double vol = height * width * depth;
        System.out.println("Volume is: " + vol);
    }
}

/* Akany this holo akta keyword. */
```

Class-2 er code:

```
package javacode;

public class Test {

    public static void main(String[] args) {

        Dhaka box1 = new Dhaka(10, 20, 30);
        Dhaka box2 = new Dhaka(40, 50, 60);

        box1.displayVolume();
        box2.displayVolume();
    }
}
```

Output:

```
Volume is: 6000.0
Volume is: 120000.0
```

```
84. Math class(pow, log, max, min, ceil, floor, sqrt, pi)
package javacode;

public class Dhaka {

    public static void main(String[] args) {

        int a = Math.abs(-2);
        System.out.println(+a);

        double d = Math.abs(-2.5);
        System.out.println(+d);

        System.out.println(Math.sqrt(25));
        System.out.println(Math.pow(2, 3));
        System.out.println(Math.PI);
        System.out.println(Math.log(2));
        System.out.println(Math.exp(1));
        System.out.println(Math.max(2.5, 5.6));
        System.out.println(Math.min(3, 4));
        System.out.println(Math.ceil(5.6));
        System.out.println(Math.floor(5.6));
    }
}
```

Abs aygula holo method

Output:

2

2.5

5.0

8.0

3.141592653589793

0.6931471805599453

2.718281828459045

5.6

3

6.0

5.0

85. Method overloading

```
package javacode;

public class Dhaka {

    void add() {
        System.out.println("Nothing to add");
    }

    void add(int a, int b) {
        System.out.println(a + b);
    }

    void add(double a, double b) {
        System.out.println(a + b);
    }

    void add(int a, int b, int c) {
        System.out.println(a + b + c);
    }

}

/*
Akoi namy method declare kora jaby. But parameter
obboisoi different hoty hoby.

*/
Same method name.
Parameter lists are different.
Inside the same class.
```

```
package javacode;

public class Test {

    public static void main(String[] args) {

        Dhaka ob = new Dhaka();
        ob.add();
        ob.add(5, 10);
        ob.add(5.5, 10.5);
        ob.add(5, 10, 20);

    }
}
```

Output:

Nothing to add

15

16.0

35

86. Automatic type conversion in java

Java's automatic type conversion

Type	Valid promotions
double	None
float	double
long	float or double
int	long, float or double
char	int, long, float or double
short	int, long, float or double
byte	short, int, long, float or double
boolean	None (boolean values are not considered to be numbers in Java)

Fig. 6.5 Allowed promotions for primitive types.

87. Debugging java code

```
package javacode;

public class Dhaka {

    public static void main(String[] args) {

        int sum = 0;
        for (int i = 1; i <= 10; i++) {
            sum = sum + i;
        }

        System.out.println("Sum is: "+sum);
    }
}
```

88. Call by value

Class-1 er code:

```
package dhaka;
```

```
public class Dhaka {
```

```
    void change(int i) {  
        i = 20;  
    }  
}
```

Class-2 er code:

```
package dhaka;
```

```
public class Test {
```

```
    public static void main(String[] args) {  
  
        Dhaka ob = new Dhaka();  
        int x = 10; //primitive data  
        System.out.println("x before call: " + x);  
  
        ob.change(x);  
        System.out.println("x after call: " + x);  
    }  
}
```

```
/*
    Actual parameter: holo x abong
    Formal parameter holo i
    Call by value te original man change hoy na.
```

```
*/
```

Output:

```
x before call: 10
x after call: 10
```

Call-by-value

- If we call a method by passing-a-value (primitive data) then it is known as call-by-value.
- The value is copied to a method parameter.
- changes to that formal parameter doesn't affect the actual parameter.
- In call-by-value original value doesn't change.

89. Call by reference

Class-1 er code:

```
package dhaka;
```

```
public class Dhaka {
```

```
    String name;
```

```
    void change(Dhaka r2) {
```

```
        r2.name = "kibria";
```

```
    }
```

```
}
```

Class-2 er code:

```
package dhaka;
```

```
public class Test {
```

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

```
        Dhaka r1 = new Dhaka();
```

```
        r1.name = "Golam";
```

```
        System.out.println("Before calling: "+r1.name);
```

```
        r1.change(r1);
```

```
        System.out.println("After calling: "+r1.name);
```

```
    }
```

```
}
```

```
Before calling: Golam
```

```
After calling: Kibria
```

Call-by-reference

- If we call a method by passing-a-reference type data (object, String etc.) then it is known as call-by-reference.
- changes to that formal parameter does affect the actual parameter.
- In call-by-reference original value gets change.

90. Variable length arguments

Class-1 er code:

```
package dhaka;
```

```
public class Dhaka {
```

```
    void add(int... num) {
        int sum = 0;
        for (int x : num) {
            sum = sum + x;
        }
        System.out.println(sum);
    }
}
```

Class-2 er code:

```
package dhaka;
```

```
public class Test {
```

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

        Dhaka ob = new Dhaka();
        ob.add(10, 20);
        ob.add(10, 20, 30);
        ob.add(10, 20, 30, 40);
        //I can do as much as i want.
    }
}
```

Main er moddy arguments thaky.

Ay program a j loop ta use korychi seta holo enhace loop.

Java te function k method boly.

Output:

30

60

100

91 Factorial using recursion

Class-1 er code:

```
package dhaka;
```

```
public class Dhaka {
```

```
    int fact(int n) {
        if (n == 1) {
            return 1;
        } else {
            return n * fact(n - 1);
        }
    }
}
```

Class-2 er code:

```
package dhaka;
```

```
public class Test {
```

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

        Dhaka ob = new Dhaka();
        int result = ob.fact(5);
        System.out.println("Factorial of 5 is: "+result);

        result = ob.fact(6);
        System.out.println("Factorial of 6 is: "+result);
    }
}
```

92. Factorial

```
package dhaka;

public class Dhaka {

    public static void main(String[] args) {

        int num = 5, fact = 1;
        for (int i = 1; i <= num; i++) {
            fact = fact * i;
        }
        System.out.println("Factorial of 5 is: " + fact);
    }
}
```

Output:

Factorial of 5 is: 120

Recursion	Iteration
1. Use selection structure (if,else,switch..)	1. Use repetition structure (for, while, do while)
2. Terminates when base case is satisfied.	2. Terminates when continuation condition fails.
3. It's slow	3. It's fast
4. Code is smaller	4. Code is bigger

93. Encapsulation

Benefits of Encapsulation

1. Provides data hiding
2. Reusability
3. Code can be modified without breaking the code.
4. Maintainability : Hiding implementation details reduces complexity.

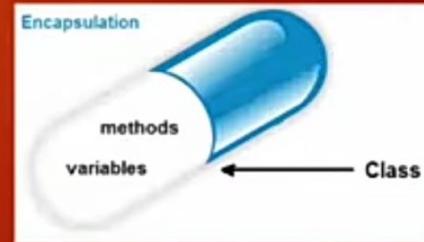
- How to do encapsulation ?

1. Declare the variables as private.
2. Provide public setter and getter method to modify and get the variables value.

Encapsulation

Encapsulation is a process of

1. Packaging Variables and methods into a single unit.



- There are 4 core concepts in OOP.



Access modifier in java

- There are four types of access modifier in java.

1. private
2. default
3. protected
4. public

private int age; //private member

int age; //default member

protected int age; //protected member

public int age; //public member

94. Setter and getter method.

Class-1 er code:

```
package dhaka;

public class Dhaka {

    private String name;
    private int age;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }
}
```

Class-2 er code:

```
package dhaka;

public class Test {

    public static void main(String[] args) {

        Dhaka p1 = new Dhaka();
        p1.setName("Golam Kibria");
        p1.setAge(20);

        System.out.println(p1.getName());
        System.out.println(p1.getAge());
    }
}
```

Output:

Name: Golam Kibria

Age: 20

95. Inheretence

Class-1 er code:

```
package dhaka;

public class Dhaka {

    String name;
    int age;

    void displayInfo1() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }
}
```

Class-2 er code:

```
package dhaka;

public class Teacher extends Dhaka {

    //name, age, displayInfo1 akany automatic choly ascy.
    String qualification;

    void displayInfo2() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Qualification: " + qualification);
    }
}
```

```
/*
    name r age ay dui ta line na likhy ami chily
    diplayInfo1() ay method taky call dity partam.
    Taholeo program thik vaby run hoto kono somosa charai.
*/
```

Class-3 er code:

```
package dhaka;
```

```
public class Test {

    public static void main(String[] args) {

        Teacher t1 = new Teacher();
        t1.name = "Golam kibria";
        t1.age = 20;
        t1.qualification = "Study in CSE";
        t1.displayInfo2();
    }
}
```

```
/*
    Chaily aro akta teacher er info add kora jaby.
    Sekhetry sudu t1 er poriborty t2 dilei hoby.
*/
```

Output:

```
Name: Golam kibria
Age: 20
Qualification: Study in CSE
```

96. Inhereting private number

Class-1 er code:

```
package dhaka;

public class Dhaka {

    private String name;
    private int age;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

}
```

Class-2 er code:

```
package dhaka;

public class Teacher extends Dhaka {

    private String qualification;

    public String getQualification() {
        return qualification;
    }

    public void setQualification(String qualification) {
        this.qualification = qualification;
    }

    /*
     * void displayInfo(){
     * System.out.println("Name: "+getName());
     * System.out.println("Age: "+getAge());
     * System.out.println("Occu: "+getQualification());
     */
}
```

```
Class-3 er code:  
package dhaka;  
  
public class Test {  
  
    public static void main(String[] args) {  
  
        Teacher t1 = new Teacher();  
        t1.setName("Golam kibria");  
        t1.setAge(20);  
        t1.setQualification("Student");  
        //t1.displayInfo();  
        System.out.println("Name: "+t1.getName());  
        System.out.println("Age: "+t1.getAge());  
        System.out.println("Occu: "+t1.getQualification());  
    }  
}  
  
/*  
 * t1.displayInfo(); Aybabu korly teahcer class a  
 * displayInfo() dity hoby.  
 */
```

Output:
Name: Golam kibria
Age: 20
Occu: Student

97. Instance of operation

```
package Java;

public class Animal {

}

package Java;

public class Person extends Animal {

}

package Java;

public class Teacher extends Person {

}

package Java;

public class Test {

    public static void main(String[] args) {

        Animal a = new Animal();
        Person p = new Person();
        Teacher t = new Teacher();

        System.out.println(a instanceof Animal);
        System.out.println(p instanceof Animal);
        System.out.println(t instanceof Person);
        System.out.println(t instanceof Animal);
        System.out.println(p instanceof Teacher);
    }
}
```

Output:

```
true
true
true
true
false
```

98. Types of inheritance

Types of Inheritance

- There are 4 types inheritance in OOP language.
- 1. Single inheritance
- 2. Multilevel inheritance
- 3. Hierarchical inheritance
- 4. Multiple Inheritance

[Click here to subscribe](#)

1. Single Inheritance



In this sort of Inheritance,
One subclass inherits from
one superclass

```
Public class A {  
    -----  
    -----  
}  
Public class B extends A {  
    -----  
    -----  
}
```

[Click here to subscribe](#)

2. Multilevel Inheritance

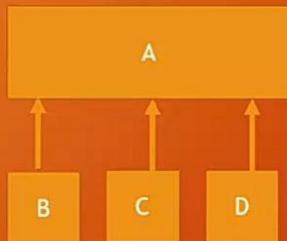


In this sort of Inheritance,
The super class for one is
the subclass for other.

```
Public class A {  
    -----  
}  
Public class B extends A {  
    -----  
}  
Public class C extends B {  
    -----  
}
```

[Click here to subscribe](#)

3. Hierarchical Inheritance

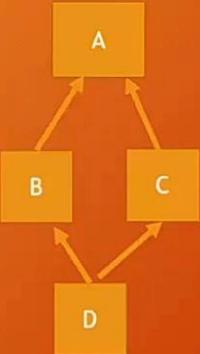


In this sort of Inheritance,
Multiple subclasses derived
from single superclass.

```
Public class A {  
    -----  
}  
Public class B extends A {  
    -----  
}  
Public class C extends A {  
    -----  
}
```

[Click here to subscribe](#)

4. Multiple Inheritance



```
Public class A {  
-----  
}  
-----  
}  
Public class B extends A {  
-----  
}  
-----  
}  
Public class C extends A {  
-----  
}  
-----  
}  
Public class D extends A,B,C {  
-----
```

[Click here to subscribe](#)

99. Method overriding

```
package Overriding;

public class Person {

    String name;
    int age;

    void displayInfo() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }
}

package Overriding;

public class Teacher extends Person {

    //name, age, displayInfo() auto choly aschy
    String qualification;

    @Override
    void displayInfo() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Occu: " + qualification);
        System.out.println("");
    }
}

package Overriding;

public class Test {

    public static void main(String[] args) {

        Teacher t = new Teacher();
        t.name = "Golam kibria";
        t.age = 20;
        t.qualification = "Student";
        t.displayInfo();

        Person p = new Person();
        p.name = "Mottaki Billah";
        p.age = 21;
        p.displayInfo();
    }
}
```


100. Difference between overloading and overriding.

Method overloading vs Method overriding

```
public class Overload {  
  
    void add(double a,double b){  
        System.out.println(a+b);  
    }  
  
    void add(int a,int b,int c){  
        System.out.println(a+b+c);  
    }  
  
    void add(){  
        System.out.println("Nothing to add");  
    }  
}
```

```
public class Person {  
    String name;  
    int age;  
  
    void displayInformation(){  
        System.out.println("Name : "+name);  
        System.out.println("Age : "+age);  
    }  
  
}  
  
public class Teacher extends Person {  
    String qualification;  
  
    @Override  
    void displayInformation(){  
        System.out.println("Name : "+name);  
        System.out.println("Age : "+name);  
        System.out.println("Qualification : "+name);  
    }  
}
```

Method overriding vs Method overloading

Method Overloading	Method Overriding
1. Parameter must be different.	1. Parameter must be different.
2. It occurs within the same class.	2. It occurs between two classes - sub class and a super class.
3. Inheritance is not involved.	3. Inheritance is involved.
4. Return type may or may not be same.	4. Return type must be same.
5. One method does not hide another.	5. child method hides parent another.

101. Super keyword-1

```
package Java;

public class A {
    int x = 10;
}

package Java;

public class B extends A {

    //int x = 10 auto choly ascy.
    int x = 5;

    void display() {
        System.out.println(x);
    }
}

/*
    Akanay x = 5 holo ay class er local variable. Jar jonno atar priority
besi tai x = 5 print hoyechy.
    Jodi x = 10 print korty chaitam tokon System.out.println(super.x) likty
hoto.
    Akany super holo akta keyword.
*/
```

```
package Java;
```

```
public class Test {

    public static void main(String[] args) {

        B ob = new B();
        ob.display();
    }
}
```

Output: 5

102. Super keyword-2

```
package Java;

public class A {

    void display(){
        System.out.println("Inside A class");
    }
}

package Java;

public class B extends A {

    @Override
    void display(){
        System.out.println("Inside B class");
    }
}

package Java;

public class Test {

    public static void main(String[] args) {

        B ob = new B();
        ob.display();
    }
}
```

Output: Inside B class

103. Super class-3

```
package Java;

public class A {
    A(){
        System.out.println("A's constructor");
    }
}

package Java;

public class B extends A {
    B(){
        //super(); na dileo problem nai. Output same e asby.
        System.out.println("B's Constructor");
    }
}

package Java;

public class Test {
    public static void main(String[] args) {
        B ob = new B();
    }
}
```

Output:

```
A's constructor
B's Constructor
```

104. Super class-4

```
package java1;

public class Vehicle {

    String color;
    double weight;

    Vehicle(String c, double w) {
        color = c;
        weight = w;
    }

    void display() {
        System.out.println("Color: " + color);
        System.out.println("Weight: " + weight);
    }
}

package java1;

public class Car extends Vehicle {

    int gear;

    Car(String c, double w, int g) {
        super(c, w);
        gear = g;
    }

    @Override
    void display() {
        super.display();
        System.out.println("Gear: " + gear);
    }
}

package java1;

public class Test {

    public static void main(String[] args) {

        Car volvo = new Car("While", 350, 5);
        volvo.display();
    }
}
```


105. This keyword-1

```
package Java;

public class A {

    String name;
    int age;
    String color;

    A(String name, int age) {
        this.name = name;
        this.age = age;
    }

    A(String name, int age, String color) {
        this(name, age);
        this.color = color;
    }

    void display() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Hair color: " + color);
    }
}

package Java;

public class Test {

    public static void main(String[] args) {
        A p1 = new A("Golam kibria", 20); //Black dileo hoto
        p1.display();

        A p2 = new A("Tausif", 20, "Black");
        p2.display();
    }
}
```

Output:

```
Name: Golam kibria
Age: 20
Hair color: null
Name: Tausif
Age: 20
Hair color: Black
```

106. This keyword-2

```
package Java;

public class A {

    void message(){
        System.out.println("I am message method");
    }

    void display(){
        message();
        System.out.println("I am display method");
    }
}
```

```
package Java;
```

```
public class Test {

    public static void main(String[] args) {
        A person = new A();
        person.display();
    }
}
```

Output:

```
I am message method
I am display method
```

107. Final keyword-2[For more info visit geekforgreeks.com]

```
package Java;

public class A {

    final String uname = "DIU";
    final int fees; //blank final variable

    A(){
        fees = 2000;
    }

    void display(){
        System.out.println("University name: "+uname);
        System.out.println("Fees: "+fees);
    }
}

/*
static final int fees;
static{
    fees = 2000;
}
aybabu likha lagtw jodi static use kortam
*/
```

```
package Java;

public class Test {

    public static void main(String[] args) {
        A ob = new A();
        ob.display();
    }
}
```

Output:

```
University name: DIU
Fees: 2000
```

108. Final keyword-3

```

package Java;

public class A {
    void display() {
        System.out.println("University information");
    }
}

package Java;

public class B extends A {
    void display2() {
        System.out.println("Students information");
    }
}

package Java;

public class Test {
    public static void main(String[] args) {
        B s1 = new B();
        s1.display();
        s1.display2();
    }
}

```

Output:
 University information
 Students information

- Kono method k final hisaby declare korly taky override kora jaby na.
- Finla class k extends kora jaby na.
- We can inherite final method but not override.
- We can initialize blank final variable using constructor.
- We initialize static blank final variable using static keyword before final keyword.



Compile Time polymorphism / static binding -> Method overloading

```
public class Overload {  
    void add(double a,double b){  
        System.out.println(a+b);  
    }  
  
    void add(int a,int b,int c){  
        System.out.println(a+b+c);  
    }  
  
    void add(){  
        System.out.println("Nothing to add");  
    }  
}  
  
public class OverloadTest {  
    public static void main(String[] args) {  
        Overload ob = new Overload();  
        ob.add();  
        ob.add(6.5, 5.5);  
        ob.add(5, 10, 20);  
    }  
}
```

In above example there are 3 versions of add methods.
The compiler looks at the method signature and decides
Which method to call at the compile time.

Click here to subscribe

Run Time polymorphism / dynamic binding -> Method overriding

```
public class Person {  
    String name;  
    int age;  
  
    void displayInformation(){  
        System.out.println("Name : "+name);  
        System.out.println("Age : "+age);  
    }  
}  
  
public class Teacher extends Person {  
    String qualification;  
  
    @Override  
    void displayInformation(){  
        System.out.println("Name : "+name);  
        System.out.println("Age : "+name);  
        System.out.println("Qualification : "+qualification);  
    }  
}
```

```
Class Test{  
    public static void main(String[] args) {  
  
        Person p = new Person();  
        p.displayInformation();  
  
        Person t = new Teacher();  
        t.displayInformation();  
    }  
}
```

Polymorphism is a mechanism where a parent class reference variable can take many forms (It can refer object from different classes.)

[Click here to subscribe](#)

110. Polymorphism-2

```
package Java;

public class A {
    void display() {
        System.out.println("I am A");
    }
}

package Java;

public class B extends A {
    //A er display function ta auto choly ascy.
    @Override
    void display() {
        System.out.println("I am B");
    }
}

package Java;

public class C extends A {
    @Override
    void display() {
        System.out.println("I am C");
    }
}

package Java;

public class Test {
    public static void main(String[] args) {
        A p = new A();
        p.display();

        p = new B();
        p.display();

        p = new C();
        p.display();
    }
}
```


111. Polymorphism-3

```
package Java;

public class Shape {

    double area(){
        System.out.print("Area of shape: ");
        return 0;
    }
}

package Java;

public class Rectangul extends Shape {

    double length, width;

    Rectangul(double length, double width) {
        this.length = length;
        this.width = width;
    }

    @Override
    double area() {
        System.out.print("Area of rectangul: ");
        return length * width;
    }
}

package Java;

public class Triangle extends Shape {

    double base, height;

    Triangle(double base, double height) {
        this.base = base;
        this.height = height;
    }

    @Override
    double area() {
        System.out.print("Area of triangle is: ");
        return 0.5 * base * height;
    }
}
```

```
package Java;

public class Test {

    public static void main(String[] args) {

        Shape s = new Shape();
        Rectangul r = new Rectangul(10, 20);
        Triangle t = new Triangle(15, 8);

        System.out.println(s.area());
        System.out.println(r.area());
        System.out.println(t.area());

        /*
            Shape[] s = new Shape[3];
            s[0] = new Shape();
            s[1] = new Rectangul(10, 20);
            s[2] = new Triangle(15, 4);
            //super class er 3 ta refer variable nilam.
            for (int i = 0; i < 3; i++) {
                System.out.println(s[i].area());
            }
        */
    }
}
```

Output:

```
Area of shape: 0.0
Area of rectangul: 200.0
Area of triangle is: 60.0
```

Abstract method

- Non abstract method :

```
void message(){  
.....  
}
```

- Abstract method :

```
abstract void message();
```

- Points to remember about abstract method.
- Abstract method has no body
- It must be ends with a semicolon
- It must be in the abstract class.
- It must be overridden.
- It can never be final and static.

[Click here to subscribe](#)

Abstract class

- Abstract class : abstract class A { }
- Abstract class can't be instantiated.
- Abstract class can have abstract and non abstract method(concrete method)
- If you extend an abstract class you have to use all its abstract methods or you have to declare the class as Abstract itself.

[Click here to subscribe](#)

- Abstraction class er object toiri kora jay na kintu refer variable toiri kora jay.
- Supper class er refer variable dara sub class er object guloky refer kora jay.

113. Abstraction-2

```
package Java;

public abstract class MobileUser {

    void call() { //non abstract
        System.out.println("Call method");
    }

    abstract void sendmessage(); //abstract
}

package Java;

public class Rahim extends MobileUser {

    @Override
    void sendmessage() {
        System.out.println("Hi, I am Rahim");
    }
}

package Java;

public class Karim extends MobileUser {

    @Override
    void sendmessage() {
        System.out.println("Hi, I am karim");
    }
}
```

```
package Java;

public class Test {

    public static void main(String[] args) {

        //super class er akta refer variable toiri kory nilam.
        MobileUser mu;

        mu = new Rahim();
        mu.sendmessage();

        mu = new Karim();
        mu.sendmessage();
    }
}
```

Output:

```
Hi, I am Rahim
Hi, I am karim
```

114. Problem solving

```
package javacode;

public abstract class Shape {

    double dim1, dim2; //dim means dimention

    Shape(double dim1, double dim2) {
        this.dim1 = dim1;
        this.dim2 = dim2;
    }

    abstract void area(); //abstract class er body thaky na.
}

package javacode;

public class Circle extends Shape {

    Circle(double r) {
        super(r, r);
    }

    //@Override
    void area() {
        double result = 3.1416 * dim1 * dim2;
        System.out.println("Circul area: " + result);
    }
}

package javacode;

public class Triangle extends Shape {

    Triangle(double dim1, double dim2) {
        super(dim1, dim2);
    }

    //@Override
    void area() {
        double result = 0.5 * dim1 * dim2;
        System.out.println("Triangle area: " + result);
    }
}
```

```
package javacode;

public class Rectangul extends Shape {

    Rectangul(double dim1, double dim2) {
        super(dim1, dim2);
    }

    // @Override
    void area() {
        double result = dim1 * dim2;
        System.out.println("Rectangul area: " + result);
    }
}
```

```
package javacode;

public class Test {

    public static void main(String[] args) {
        Shape sp;

        sp = new Rectangul(10, 20);
        sp.area();

        sp = new Triangle(10, 20);
        sp.area();

        sp = new Circle(5);
        sp.area();
    }
}
```

Output:

```
Rectangul area: 200.0
Triangle area: 100.0
Circul area: 78.54
```

115. Interface

```
package Java;

public interface Animal { //ata kintu interface class na.

    public abstract void eat();
}

package Java;

public class Dog implements Animal {

    //@Override
    public void eat() {
        System.out.println("Dogs can eat");
    }
}

package Java;

public class Cat implements Animal {

    //@Override
    public void eat() {
        System.out.println("Cat can eat fish");
    }
}

package Java;

public class Test {

    public static void main(String[] args) {

        Dog d = new Dog();
        d.eat();

        Cat c = new Cat();
        c.eat();
    }
}

Output:
Dogs can eat
Cat can eat fish
```

interface

- What is an Interface?
- An interface is a collection of abstract methods.

[Click here to subscribe](#)

- Why do we need an Interface ?
 1. For fully abstraction.
 2. It supports multiple inheritance.

[Click here to subscribe](#)

- Interfaces have the following properties
- - Each method in an interface is also implicitly abstract, so the abstract keyword is not needed.
 - Methods in an interface are implicitly public.
 - A class can inherit from just **one** superclass, but can implement **multiple** interfaces!

[Click here to subscribe](#)

By default protita method e public interface er moddy thaky.

Interface & Abstract er object toiri kora jay na.

116. Why interface support multiple inheritent

```
package Java;

public interface A {
    void play();
}

package Java;

public interface B {
    void play();
}

package Java;

public class C implements A, B {
    public void play() {
        System.out.println("Hello I am from C");
    }
}

package Java;

public class Test {

    public static void main(String[] args) {
        C ob = new C();
        ob.play();
    }
}
```

Output:
Hello I am from C

117. Interface vs Abstract

- How interface is similar to a class.
- 1. Interface can have any number of methods.
- 2. It has same file extension as class (.java)

[Click here to subscribe](#)

- How interface is different from a class ?
- 1. You can't instantiate an interface.
- 2. Interface doesn't contain constructor.
- 3. All the methods in interface are abstract.
- 4. An interface can't have instance variables.
- 5. An interface can extend multiple interface.

[Click here to subscribe](#)

Interface vs Abstract class

Interface	Abstract class
1. can only have abstract methods.	1. Can have abstract and non abstract methods.
2. It supports multiple inheritance.	2. It doesn't supports multiple inheritance.
3. Can only have static and final variable.	3. Can have static, non static, final and non final variable.
4. Fully abstract.	4. Partial abstraction.
5. Example	5. Example
<pre>interface Animal{ void eat(); }</pre>	<pre>abstract class Animal{ abstract void eat(); }</pre>

[Click here to subscribe](#)

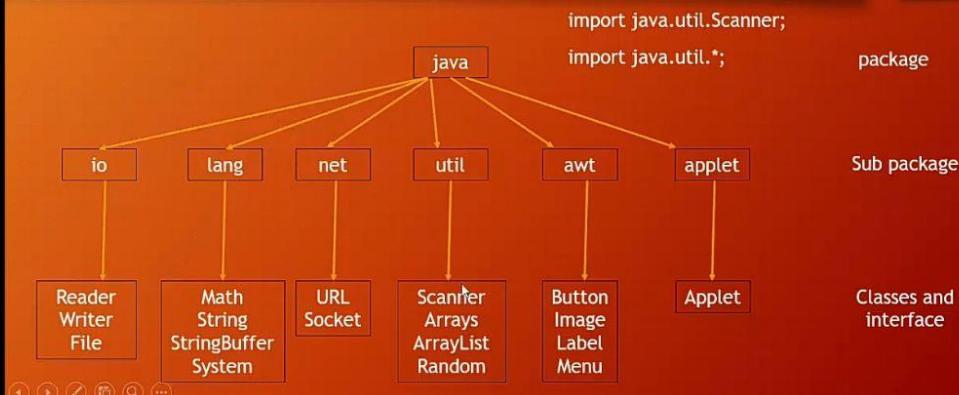
118. Package

- A package is a group of related classes, interfaces and sub packages.

- There are 2 types of package in java.



Built-in-package



119. Package

```
/*PACKAGE-11*/
package java1;

public class Cat {

    public static void main(String[] args) {

        //Dog d = new Dog();
        Java2.Dog d = new Java2.Dog();
    }
}

package java1;

import Java2.*;

public class Dog {

    public static void main(String[] args) {

        Cat c = new Cat(); //ata java1 er moddei asy
        Cow cow = new Cow();
        Donkey donk = new Donkey();
    }
}

/*PACKAGE-2*/
package Java2;

public class Cow {

}

package Java2;

public class Dog {

    public static void main(String[] args) {

    }
}

package Java2;

public class Donkey {

}
```

120. Access modifier

```
/*PACKAGE-1*/
package JavaCode1;

public class A {

    public void display() {
        System.out.println("Hi");
    }
}

package JavaCode1;

public class Test {

    public static void main(String[] args) {
        A ob = new A();
        ob.display();
    }
}

/*PACKAGE-2*/
package JavaCode2;

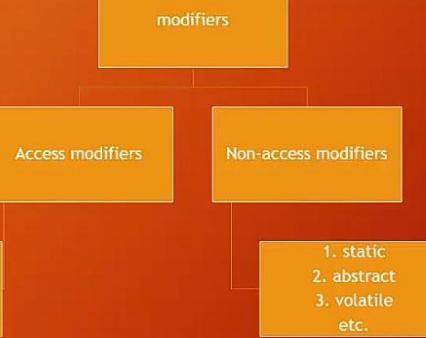
import JavaCode1.A;

public class C extends A {

    public static void main(String[] args) {

        A ob = new A();
        ob.display();
    }
}
//extends A na dileo output same asby.
```

Output: Hi



```
class A {  
    int x;  
}  
  
class A {  
    private int x;  
}
```

[Click here to subscribe](#)



121. Type casting

```
package java;

public class TypeCasting {

    public static void main(String[] args) {

        double d = 10.5;
        int x = (int) d;

        System.out.println(x);
    }
}
Output: 10

/*NICHER 3 TA CLASS AKTA CODE*/
package java;

public class Person {

    void display(){
        System.out.println("Person class");
    }
}

package java;

public class Teacher extends Person{

    @Override
    void display(){
        System.out.println("Teacher class");
    }
}

package java;
public class Test {

    public static void main(String[] args) {

        /*Akta sub class er object teacher k super class er object perosn a
raklam*/
        Person p = new Teacher(); //upcasting
        p.display();

        Teacher t = (Teacher) new Person();
        t.display();
    }
}
```

Output:

Teacher **class**

porer ta print hoby na karon oitay compiler error na holeo runtime error hoy.

Type casting

- What is Type casting ?

-> Converting one data type to another is called type casting.

byte → short → int → long → float → double
→
widening

double → float → long → int → short → byte
→
Narrowing

[Click here to subscribe](#)

122. Anonimous class

```
package Java;

public class Person {

    void display() {
        System.out.println("Person class");
    }
}

package Java;

public class Test {

    public static void main(String[] args) {

        Person p = new Person() {

            @Override
            void display() {
                System.out.println("Test class");
            }
        };
        p.display();
    }
}
```

Output: Test class

123. Exception handling

- What is an Exception?

-> An exception is a run time error.

-> An exception is an abnormal condition that arises in a code sequence at run time.

```
statement 1;  
statement 2;  
statement 3;//exception occurs  
statement 4;  
statement 5;  
statement 6;  
statement 7;
```

```
int x = 10;  
int y = 0;  
int result = x/y; //exception occurs here
```

Different types of exception

```
int x = 10;  
int y = 0;  
int result = x/y; → ArithmeticException : Can't divide a number by 0
```

```
String name =null;  
System.out.println(name.charAt(0)); → NullPointerException
```

```
String name ="Anisul"; // length is 6  
System.out.println(name.charAt(9)); → StringIndexOutOfBoundsException
```

Different types of exception

```
int num = Integer.parseInt ("anis") ; -----> NumberFormat Exception
```

```
File file = new File ("D://file.txt") ; -----> FileNotFoundException
```

```
int a[ ] = new int[5];  
a[5] = 32; -----> ArrayIndexOutOfBoundsException
```

Some other exceptions

ClassNotFoundException
IOException
NoSuchMethodException
Etc.

- What is exception handling?
-> The **exception handling** is one of the powerful *mechanism to handle the runtime errors* .

- Exception handling is managed by 5 keywords.
 1. try
 2. catch
 3. finally
 4. throw
 5. throws

Try-catch-finally block

```
try{  
    //code you want to monitor  
} catch (ExceptionType1 e1) {  
    //exception handler for exception  
} catch (ExceptionType2 e2) {  
    //exception handler for exception  
}  
.....  
finally{  
    //block of code to be executed after try block  
}
```

124. Exception handling

```
package JavaCode;

public class Java {

    public static void main(String[] args) {

        try {
            int x = 10, y = 5;
            int result = x / y;
            System.out.println("Result: " + result);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Exception: " + e);
        } catch (ArithmeticException a1) {
            System.out.println("Exception: " + a1);
        } catch (Exception ex) {
            System.out.println("Exception: " + ex);
        } finally {
            System.out.println("Last line of the program");
        }
    }
}
/*Check the program using different output*/
```

Output:

```
Result: 2
Last line of the program
```

Try block a jeita through korby catch block seta receive korby.

```
package JavaCode;

public class Java {

    public static void main(String[] args) {

        try {
            int[] a = new int[4];
            a[4] = 10;
        } catch (ArithmeticException a1) {
            System.out.println("Exception: " + a1);
        } catch (Exception ex) {
            System.out.println("Exception: " + ex);
        } finally {
            System.out.println("Last line of the program");
        }
    }
}
Output:
Exception: java.lang.ArrayIndexOutOfBoundsException: 4
Last line of the program
```

Try block jeta through korby catch block seta receive korby.

125. Exception handling problem solve

```
package JavaCode;

import java.util.Scanner;

public class Java {

    public static void main(String[] args) {

        //int count = 1;
        //while(true) er poriborty
        //do{
        while (true) {
            try {
                Scanner input = new Scanner(System.in);
                System.out.println("Please enter num1: ");
                int num1 = input.nextInt();
                System.out.println("Please enter num2: ");
                int num2 = input.nextInt();

                int result = num1 / num2;
                System.out.println("Result: " + num1 + "/" + num2 + " = " +
result);
                //count = 2;
            } catch (Exception e) {
                System.out.println("Exception: " + e);
                System.out.println("You must inter integer. Please try
again.");
            }
        } //while(count == 1)
    }
}

/*
    Jodi loop ta thamaty chai sekhetry code a comment er line gulo add
kory dity hoby.
*/
```

126. Decimal format

```
package JavaCode;

import java.text.DecimalFormat;

public class Java {

    public static void main(String[] args) {

        double d = 3.12445446;
        System.out.printf("d = %.2f\n", d);

        DecimalFormat ob = new DecimalFormat("0.00");
        double x = 4.12343254;
        System.out.println("x = " + ob.format(x));
    }
}
```

Output:

```
d = 3.12
x = 4.12
```

DecimalFormat class er akta method er nam holo Format

127. `toString` method

```
package JavaCode;

public class Person {

    String name;
    int age;

    Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    @Override
    public String toString() {
        return "Name: " + name + "\nAge: " + age;
    }
}
```

```
package JavaCode;

public class Test {

    public static void main(String[] args) {

        Person p1 = new Person("Golam kibria", 20);
        Person p2 = new Person("Saim Islam", 20);

        System.out.println(p1); //toString
        System.out.println(p2);
    }
}
```

Output:

```
Name: Golam kibria
Age: 20
Name: Saim Islam
Age: 20
```

128. String comparison

```
/*EQUALS METHOD*/
package JavaCode;

public class Java {

    public static void main(String[] args) {

        String password1 = "kibria123";
        String password2 = "kibria123";
        String password3 = new String("kibria123");
        String password4 = new String("kibria123");

        System.out.println(password1.equals(password2));
        System.out.println(password1.equals(password3));
        System.out.println(password3.equals(password4));

    }
}

/*
equals() method kaj kory content niye.
Complare to the original content of the string.
*/
OUTPUT:
true
true
true
```

```

package JavaCode;

public class Java {

    public static void main(String[] args) {

        String password1 = "kibria123";
        String password2 = "kibria123";
        String password3 = new String("kibria123");
        String password4 = new String("kibria123");

        System.out.println(password1 == password2);
        System.out.println(password1 == password3);
        System.out.println(password3 == password4);
        System.out.println(password1.equalsIgnoreCase(password2));
        //jodi upper and lowercase a problem na korty chai taholy
        equalsIgnore method use korty hoby.
    }
}

/*
>equals operator kaj kory reference niye.
Akany new operator er maddomy new referance toiri korychi jar jonno
false hoicy.
*/

```

OUTPUT:

```

true
false
false
true

```

Main difference between “==” and “equals” in java is that “==” is used to compare primitive while “equals()” method is recommended to check equality of object.

129. Linked list-1

```
package JavaCode;

import java.util.LinkedList;

public class Java {

    public static void main(String[] args) {

        LinkedList<String> ob = new LinkedList<String>();

        ob.add("Afganistan");
        ob.add("Bangladesh");
        ob.add("Pakistan");
        ob.add("China");
        ob.add("Germany");

        System.out.println(ob);
    }
}
```

Output:

```
[Afganistan, Bangladesh, Pakistan, China, Germany]
```

ArrayList

3

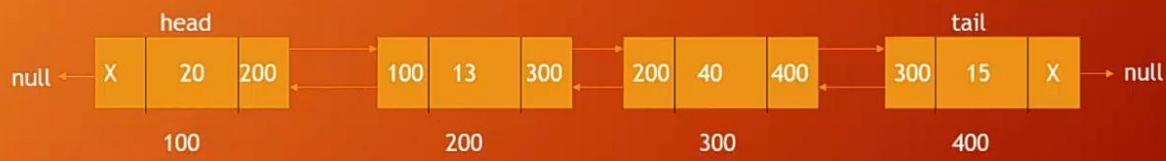
0	1	2	3
20	13	40	15

1. ArrayList class uses a dynamic array for storing the elements.
2. ArrayList is better for sorting and accessing data
3. Slow for manipulating data (deleting or inserting data)
4. can contain duplicate elements.

[Click here to subscribe](#)

LinkedList

4



1. LinkedList class uses doubly linked list to store the elements.
2. manipulating of data is fast here (deleting or inserting data)
4. can contain duplicate elements.

[Click here to subscribe](#)

130. Linked list-2

```
package JavaCode;

import java.util.LinkedList;

public class Java {

    public static void main(String[] args) {

        LinkedList<String> ob = new LinkedList<String>();

        ob.add("Afghanistan");
        ob.add("Bangladesh");
        ob.add("Pakistan");
        ob.add("China");
        ob.add("Germany");
        ob.add(5, "Brazil");
        ob.addFirst("Australia");
        ob.addLast("Japan");
        //ob.remove("Japan");
        //ob.remove(2);
        //ob.removeFirst();
        //ob.removeLast();

        for (String x : ob) {
            System.out.println(x);
        }

        System.out.println("Size of the linked list: " + ob.size());
        System.out.println("First element: " + ob.getFirst());
        System.out.println("Last element: " + ob.getLast());

        ob.clear();
        System.out.println(ob);
    }
}
```


131. Creating a student list

```
package javacode;

public class Student {

    String name, className;
    int id;

    Student(int id, String name, String className) {
        this.id = id;
        this.name = name;
        this.className = className;
    }
}

package javacode;

import java.util.LinkedList;

public class List {

    public static void main(String[] args) {

        //student linked list
        LinkedList<Student> ob = new LinkedList<Student>();

        //student create
        Student s1 = new Student(10, "kibria", "Ten");
        Student s2 = new Student(11, "Sakib", "Nine");
        Student s3 = new Student(12, "Nurul", "Eight");
        Student s4 = new Student(13, "Sajib", "Seven");

        //adding student to the studentlist
        ob.add(s1);
        ob.add(s2);
        ob.add(s3);
        ob.add(s4);

        //information display
        for (Student x : ob) {
            System.out.println(x.id + " " + x.name + " " + x.className);
        }
    }
}
```


132. HashMap

```
package javacode;

import java.util.HashMap;

public class Map {

    public static void main(String[] args) {

        HashMap<Integer, String> ob = new HashMap<Integer, String>();

        ob.put(10, "kibria");
        ob.put(11, "Sakib");
        ob.put(12, "Mottaki");

        System.out.println(ob.get(10));
        System.out.println(ob.get(11));
        System.out.println(ob.get(12));
    }
}
```

Output:
kibria
Sakib
Mottaki

133. HashSet

```
package javacode;

import java.util.HashSet;

public class Map {

    public static void main(String[] args) {

        HashSet<String> name = new HashSet<String>();

        name.add("Apple");
        name.add("Banana");
        name.add("Orange");
        name.add("Strawberry");

        System.out.println(name);
        System.out.println("Size of hashset: " + name.size());

        name.remove("Apple");
        System.out.println(name);
        name.clear();
        System.out.println(name);
        boolean value = name.isEmpty();
        System.out.println(value);
    }
}
```

OUTPUT:

```
[Apple, Strawberry, Orange, Banana]
Size of hashset: 4
[Strawberry, Orange, Banana]
[]
true
```

134. How to create a file and folder

```
package file;

import java.io.File;

public class FileCode {

    public static void main(String[] args) {

        //kibaby folder toiri korty korty hoy:
        File dir = new File("Daffodil");
        //File ob = new File("C:/Users/golam/OneDrive/Desktop//Daffodil");
        //aybabu korly display te akta folder create hoy.
        dir.mkdir(); //directory/folder will be created
        //ay dui line korar por code ta run korly project er location a
        //daffodil nam a akta folder create hoy.
        String location = dir.getAbsolutePath();
        //ata dara file ta kon folder a toiri hoicy seta jana jay
        System.out.println(location);
        System.out.println(dir.getName());
        //ata dara folder er nam ta janty parbo

        if (dir.delete()) {
            System.out.println(dir.getName() + " folder has been deleted");
            //ay line ta execute hoy package Location hoty daffodil file ta
        delete hoye jaby
    }
}
```

```
//kibaby folder er moddy file toiri korty hoy
File dir = new File("Daffodil");
dir.mkdir();
String path = dir.getAbsolutePath();

File file1 = new File(path + "/Student.txt");
File file2 = new File(path + "/Teacher.txt");
//orthat daffodil folder er moddy duita file create hoby.

try {
    file1.createNewFile();
    file2.createNewFile();
    System.out.println("Files are created");
} catch (Exception e) {
    System.out.println(e);
}

file2.delete();
if (file2.exists()) {
    System.out.println("File exists");
} else {
    System.out.println("File does not exists");
}
}

}
```

135. How to write within a file

```
package file;

import java.io.FileNotFoundException;
import java.util.Formatter;
import java.util.Scanner;

class CodeFile {

    public static void main(String[] args) {

        String name, id;
        //try catch er moddy rakty hoby. karon onk somoy oi location a file ta nao thakty pary.
        try {
            Formatter f = new
            Formatter("C:/Users/golam/Documents/File/Student.txt");

            Scanner input = new Scanner(System.in);
            System.out.print("How many numbers? ");
            int num = input.nextInt();

            for (int i = 1; i <= num; i++) {
                System.out.print("Enter students id and name: ");
                id = input.next();
                name = input.next();
                f.format("%s %s\r\n", id, name);
            }

            //user hoty input na nily nicher tin line korbo
            //f.format("%s %s\r\n", "10", "Kibria");
            //f.format("%s %s\r\n", "11", "Abid");
            //f.format("%s %s\r\n", "12", "Tausif");
            f.close();
        } catch (FileNotFoundException e) {
            System.out.println(e);
        }
    }
}
```


136. How to read a file

```
package file;

import java.io.File;
import java.util.Scanner;

public class FileRead {

    public static void main(String[] args) {

        try {
            File f = new File("C:/Users/golam/Documents/File/Student.txt");
            Scanner scan = new Scanner(f);

            while (scan.hasNext()) {
                String id = scan.next();
                String name = scan.next();
                System.out.println("ID: " + id + " Name: " + name);
            }
            scan.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

OUTPUT:

```
//file er moddy ay data gulo thakby abong console a aygulo print hoby
ID: 1 Name: asif
ID: 2 Name: kibria
ID: 3 Name: abid
```

137.1 Remove an element from an array

```
package javaapplication31;

import java.util.Arrays;
import java.util.Scanner;

public class DeleteanElement {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.print("How many elements? ");
        int n = input.nextInt();
        int[] arr = new int[n];
        for (int i = 0; i < n; i++) {
            arr[i] = input.nextInt();
        }

        System.out.println("Original Array: " + Arrays.toString(arr));

        int index = 2;

        System.out.println("Index to be removed: " + index);

        if (arr == null || index < 0 || index >= arr.length) {

            System.out.println("Original Array: " + Arrays.toString(arr));

        }

        int[] copy = new int[arr.length - 1];

        for (int i = 0, k = 0; i < arr.length; i++) {

            if (i == index) {
                continue;
            }

            copy[k++] = arr[i];
        }

        System.out.println("Resultant Array: " + Arrays.toString(copy));
    }
}
```

OUTPUT:

```
How many elements? 5
12 23 34 56 67
Original Array: [12, 23, 34, 56, 67]
Index to be removed: 2
Resultant Array: [12, 23, 56, 67]
```

137.2 Remove an element from an array using method/function

```
package javaapplication31;

import java.util.Arrays;

public class DeleteanElement {

    // Function to remove the element
    public static int[] removeTheElement(int[] arr, int index) {

        // If the array is empty
        // or the index is not in array range
        // return the original array
        if (arr == null || index < 0
            || index >= arr.length) {

            return arr;
        }

        // Create another array of size one less
        int[] copy = new int[arr.length - 1];

        // Copy the elements except the index
        // from original array to the other array
        for (int i = 0, k = 0; i < arr.length; i++) {

            // if the index is
            // the removal element index
            if (i == index) {
                continue;
            }

            // if the index is not
            // the removal element index
            copy[k++] = arr[i];
        }

        // return the resultant array
        return copy;
    }
}
```

```
// Driver Code
public static void main(String[] args) {

    // Get the array
    int[] arr = {1, 2, 3, 4, 5};

    // Print the resultant array
    System.out.println("Original Array: " + Arrays.toString(arr));

    // Get the specific index
    int index = 2;

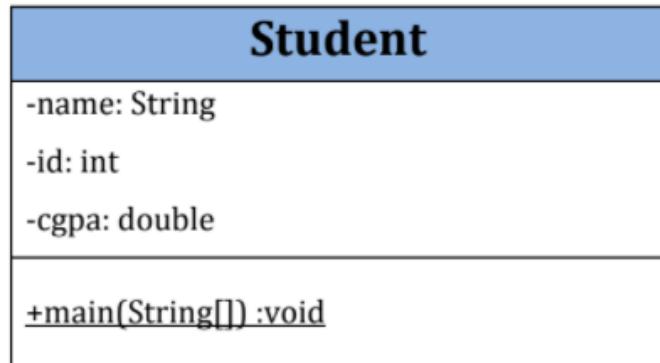
    // Print the index
    System.out.println("Index to be removed: " + index);

    // Remove the element
    arr = removeTheElement(arr, index);

    // Print the resultant array
    System.out.println("Resultant Array: " + Arrays.toString(arr));
}
}
```

UML-1

Create the **Student** Class. Then create two **objects** of this class. Take User input for the instance variables.



```
/*UML-1*/
package Labfour;

import java.util.Scanner;

public class Student {

    String name;
    int id;
    double cgpa;

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.println("Please enter s1 name, id and cgpa: ");
        Student s1 = new Student();
        s1.name = input.nextLine();
        s1.id = input.nextInt();
        s1.cgpa = input.nextDouble();

        System.out.println("Name: " + s1.name);
        System.out.println("ID: " + s1.id);
        System.out.println("CGPA: " + s1.cgpa);

        input.nextLine();

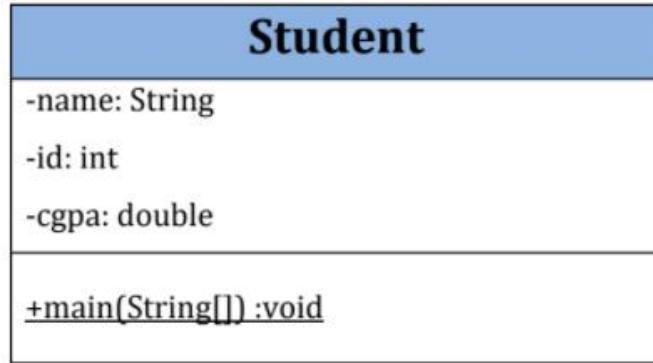
        System.out.println("\nPlease enter s2 name, id and cgpa: ");
        Student s2 = new Student();
        s2.name = input.nextLine();
        s2.id = input.nextInt();
        s2.cgpa = input.nextDouble();

        System.out.println("Name: " + s2.name);
        System.out.println("ID: " + s2.id);
        System.out.println("CGPA: " + s2.cgpa);

    }
}
```

UML-2

Create the **Student** Class. Then create **N Number objects** of this class. Take User input for the instance variables.



```
/*UML-2*/
package Labfour;

import java.util.Scanner;

public class Person2 {

    String name;
    int age;
    double cgpa;

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        Person2[] person = new Person2[5];
        System.out.print("How many students? ");
        int n = input.nextInt();

        for (int i = 0; i < n; i++) {
            person[i] = new Person2();
            System.out.print("Enter name: ");
            input.nextLine();
            person[i].name = input.nextLine();
            System.out.print("Enter age: ");
            person[i].age = input.nextInt();
            System.out.print("Enter CGPA: ");
            person[i].cgpa = input.nextDouble();
        }

        for (int i = 0; i < n; i++) {
            System.out.println("\nInformation of person: " + (i + 1));
            System.out.println("Name: " + person[i].name);
            System.out.println("Age: " + person[i].age);
            System.out.println("CGPA: " + person[i].cgpa);
        }
    }
}
```

UML-3

Create the **Student** Class. Then create two **objects** of this class. Take User input for the instance variables.

Student	
-name:	String
-id:	int
-cgpa:	double
+insertRecord	(String , int , double) : void
+displayRecord()	:void
+main	(String[]) :void

```
/*UML-3*/
package Labfour;

public class Students {

    String name;
    int id;
    double cgpa;

    void insertRecord(String name, int id, double cgpa) {
        this.name = name;
        this.id = id;
        this.cgpa = cgpa;
    }

    void displayRecord() {
        System.out.println("Name: " + name);
        System.out.println("ID: " + id);
        System.out.println("CGPA: " + cgpa);
    }
}
package Labfour;

import java.util.Scanner;

public class TestJava {

    public static void main(String[] args) {

        Students s = new Students();
        Scanner input = new Scanner(System.in);

        System.out.print("Enter name: ");
        String name = input.nextLine();
        System.out.print("Enter ID: ");
        int code = input.nextInt();
        System.out.print("Enter CGPA: ");
        double cg = input.nextDouble();

        s.insertRecord(name, code, cg);
        s.displayRecord();
    }
}
```

UML-4

Create the **Student** Class. Then create N number of **objects** of this class. Take User input for the number of objects and the instance variables [**Array of Objects**].

Student	
-name:	String
-id:	int
-cgpa:	double
+insertRecord (String , String , double) :	void
+displayRecord():	void
+main(String[])	:void

```

/*UML-4*/
package Labfour;

public class Students {

    String name;
    int id;
    double cgpa;

    void insertRecord(String name, int id, double cgpa) {
        this.name = name;
        this.id = id;
        this.cgpa = cgpa;
    }

    void displayRecord() {
        System.out.println("Name: " + name);
        System.out.println("ID: " + id);
        System.out.println("CGPA: " + cgpa);
    }
}
package Labfour;

import java.util.Scanner;

public class TestJava {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        int n = input.nextInt();
        Students[] s = new Students[n];

        for (int i = 0; i < n; i++) {
            s[i] = new Students();
            System.out.print("Enter name: ");
            input.nextLine();
            s[i].name = input.nextLine();
            System.out.print("Enter age: ");
            s[i].id = input.nextInt();
            System.out.print("Enter CGPA: ");
            s[i].cgpa = input.nextDouble();
        }

        for (int i = 0; i < n; i++) {
            s[i].insertRecord(s[i].name, s[i].id, s[i].cgpa);
            s[i].displayRecord();
        }
    }
}

```

UML-5

Suppose you are going to start your own company. You need to recruit some employees. Now, create a class named “**Employee**” to store the information of them. **Convert** the following **UML** into **Java code** and **store** information of three employees.

Employee	
-name:	String
-id:	String
-age:	int
-salary:	double
+setEmpName	(String ,String) : void
+setEmpAge	(int) : void
+setEmpSalary	(double) : void
+display_Info	() : void
+main	(String) : void

```
/*UML-5*/
package labfour;

public class Employee {

    String name;
    String id;
    int age;
    double salary;

    void setEmpNameId(String name, String id) {
        this.name = name;
        this.id = id;
    }

    void setEmpAge(int age) {
        this.age = age;
    }

    void setEmpSalary(double salary) {
        this.salary = salary;
    }

    void displayifo() {
        System.out.println("Name: " + name);
        System.out.println("ID: " + id);
        System.out.println("Age: " + age);
        System.out.println("Salary: " + salary);
    }
}
package labfour;

public class TestJava {

    public static void main(String[] args) {

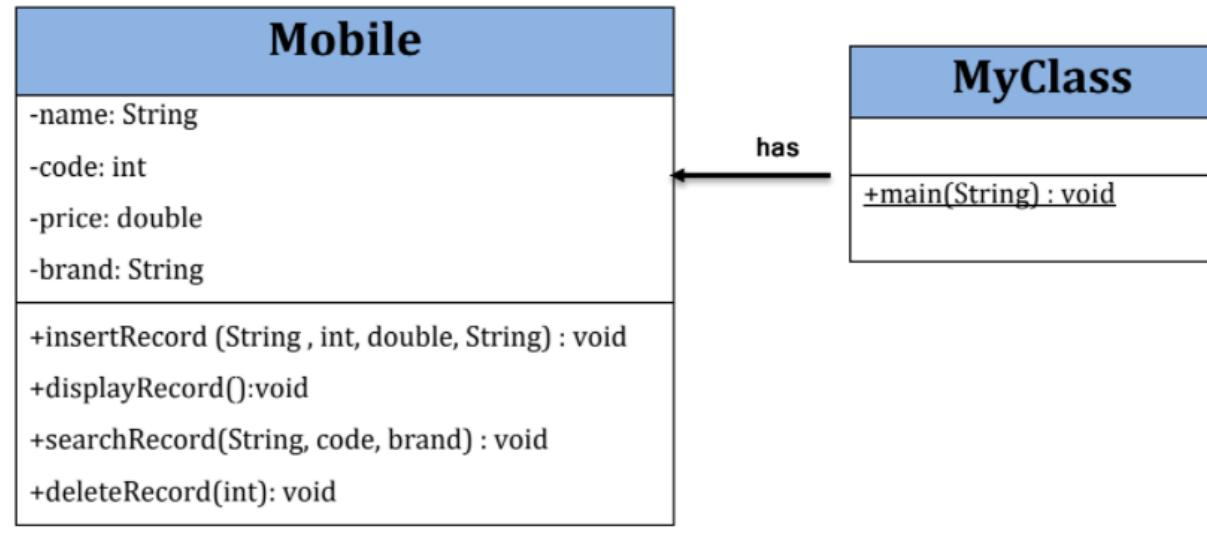
        Employee e = new Employee();

        e.setEmpNameId("Rahim", "100");
        e.setEmpAge(20);
        e.setEmpSalary(1500.25);

        e.displayifo();
    }
}
```

Convert the above UML into Java Code.

- **Store details of Mobiles.**
- **Display all the Mobile List.**
- **Search for a friend by their Name/Code/Brand.**
- You can **edit or delete** any their information.



```
/*UML-6*/
package student;

import java.util.ArrayList;
import java.util.Scanner;

public class Mobile {

    int code;
    double price;
    String name, brand;

    /*INSERT*/
    public void insertValue(int code, double price, String name, String
brand) {
        this.code = code;
        this.price = price;
        this.name = name;
        this.brand = brand;
    }

    /*DISPLAY*/
    public void displayValue() {
        System.out.println("Phone code: " + this.code);
        System.out.println("Phone price: " + this.price);
        System.out.println("Phone name: " + this.name);
        System.out.println("Phone brand: " + this.brand);

    }

    /*CREATE*/
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("How many times the loop will be executed? ");
        int n = input.nextInt();
        input.nextLine();

        Mobile[] m = new Mobile[n];
        for (int i = 0; i < n; i++) {
            m[i] = new Mobile();

            System.out.print("Enter the code: ");
            int phoneCode = input.nextInt();
            System.out.print("Enter the price: ");
            double phonePrice = input.nextDouble();
            input.nextLine();
            System.out.print("Enter the phone name: ");
            String phoneName = input.nextLine();
            System.out.print("Enter the brand name: ");
            String phoneBrand = input.nextLine();
        }
    }
}
```

```
//input.nextLine();
m[i].insertValue(phoneCode, phonePrice, phoneName, phoneBrand);
}

System.out.println("Information printing: ");
for (int i = 0; i < n; i++) {
    m[i].displayValue();
}

/*SEARCHING NAME*/
System.out.print("Enter the brand name you want to search: ");
String searchBrandName = input.nextLine();
int count = 0;
for (int i = 0; i < n; i++) {
    if (m[i].brand.equalsIgnoreCase(searchBrandName)) {
        count = 1;
    }
}
if (count == 1) {
    System.out.println("Brand name is found");
} else {
    System.out.println("Brand name is not found");
}

/*SEARCHING CODE*/
System.out.print("Enter the phone code you want to search: ");
int pCode = input.nextInt();
int c = 0;

for (int i = 0; i < n; i++) {
    if (m[i].code == pCode) {
        c = 1;
    }
}
if (c == 1) {
    System.out.println("Phone code is found");
} else {
    System.out.println("Phone code is not found");
}
```

```
/*REMOVING CODE AND NAME*/
ArrayList<Integer> r = new ArrayList<>();
ArrayList<String> s = new ArrayList<>();

System.out.print("Enter the code you want to remove? ");
int removeCode = input.nextInt();
r.add(removeCode);
//System.out.println(r);
r.remove(0);
System.out.println(r);
System.out.println("Code number Removed successfully");

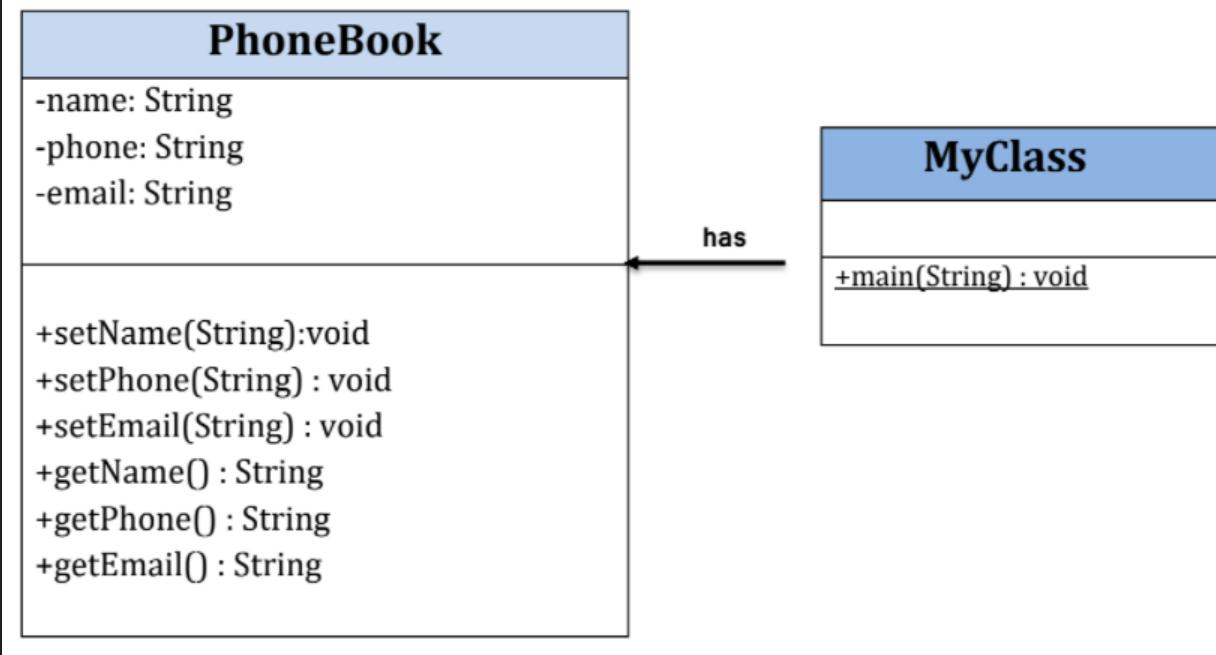
input.nextLine();
System.out.print("Enter the brand name you want ot remove? ");
String removeBrand = input.nextLine();
s.add(removeBrand);
s.remove(0);
System.out.println(s);
System.out.println("Brand Name Removed Successfully");
}

}
```

Suppose you are going to design a **PhoneBook** to store **contact details** of your **friends**.

You can -

- **Store Name, Phone Number and Email Address** of your Friends.
- **Display all** the Friend List.
- **Search for** a friend by their **Name/Phone/Email**.
- You can **edit** or **delete** any their information.



```
package labfive;

public class PhoneBook {

    String name, phone, email;

    public String getName() {
        return name;
    }

    public String getPhone() {
        return phone;
    }

    public String getEmail() {
        return email;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setPhone(String phone) {
        this.phone = phone;
    }

    public void setEmail(String email) {
        this.email = email;
    }
}
```

```
package labfive;

public class TestTwo {

    public static void main(String[] args) {

        PhoneBook p = new PhoneBook();
        p.setName("Golam kibria");
        p.setPhone("01790037447");
        p.setEmail("kibria14-522@diu.edu.bd");

        System.out.println(p.getName());
        System.out.println(p.getPhone());
        System.out.println(p.getEmail());
    }
}
```

OOP: Lab Task – 1

Display Some information in Java

1. Print “**Hello Class**” in JAVA.
2. Print **Multiple** Lines.
3. **Add** two integers and **print** the result.

Home Work

1. Print Your **Information** (Name, ID, Address, department, favorite game etc).
2. Initialize a variable with your **name** and then **print** it.
3. Find the **area** of a **Circle**

The image shows two instances of the NetBeans IDE running side-by-side, each displaying a Java application's code and its corresponding output window.

Top Window (NetBeans IDE 8.2):

- Project:** JavaApplication1
- Code Editor:** Shows the `Sum.java` file with the following code:

```
1 package javaapplication1;
2
3 import java.util.Scanner;
4
5 public class Sum {
6
7     public static void main(String[] args) {
8         Scanner input = new Scanner(System.in);
9
10        int a, b;
11        a = input.nextInt();
12        b = input.nextInt();
13        int c = a + b;
14        System.out.println("The sum is: "+c);
15    }
16 }
17 }
```
- Output Window:** Shows the application's output:

```
C:\>
Name is: Golam kibria
Age is: 20
Gpa is: 3.5
Section is: CSE
Section is: B
BUILD SUCCESSFUL (total time: 0 seconds)
```

Bottom Window (NetBeans IDE 8.2):

- Project:** JavaApplication1
- Code Editor:** Shows the `Homework.java` file with the following code:

```
1 /* 3. Find the area of a Circle */
2 package javaapplication1;
3
4 import java.util.Scanner;
5
6 public class Homework {
7
8     public static void main(String[] args) {
9
10        Scanner input = new Scanner(System.in);
11
12        double r, area;
13        System.out.print("Enter the radius: ");
14        r = input.nextDouble();
15
16        area = 3.1416 * r * r;
17
18        System.out.println("The area of circle is: " + area);
19
20    }
21 }
```
- Output Window:** Shows the application's output:

```
Enter the radius: 3.5
The area of circle is: 38.4846
BUILD SUCCESSFUL (total time: 5 seconds)
```

The image displays two side-by-side Java code editors, likely from an IDE like Eclipse or IntelliJ IDEA, running on a dark-themed interface.

Top Editor:

```
1 /* 2. Initialize a variable with your name and then print it. */
2
3 package javaapplication1;
4
5 public class Homework {
6
7     public static void main(String[] args) {
8
9         int kibria = 10;
10        System.out.println("The value of kibria is = "+kibria);
11    }
12 }
```

Bottom Editor:

```
1 /* 1. Printing My Information (Name, ID, Address, department, favorite
2 game etc. )*/
3
4 package javaapplication1;
5
6 public class Homework {
7
8     public static void main(String[] args) {
9         String name = "Golam kibria";
10        String id = "203-15-14522";
11        String add = "Keranigong-Dhaka";
12        String dpt = "CSE";
13        String game = "Angry Birds";
14
15        System.out.println("My name is: " + name);
16        System.out.println("My id is: " + id);
17        System.out.println("My address is: " + add);
18        System.out.println("My department is: " + dpt);
19        System.out.println("My favourite game is: " + game);
20    }
21 }
```

Output Window (Top):

```
run:
The value of kibria is = 10
BUILD SUCCESSFUL (total time: 0 seconds)
```

Output Window (Bottom):

```
run:
My name is: Golam kibria
My id is: 203-15-14522
My address is: Keranigong-Dhaka
My department is: CSE
My favourite game is: Angry Birds
BUILD SUCCESSFUL (total time: 0 seconds)
```

146. Lab Task-2 solve

Lab Task – 2

Taking User Input using Scanner class (**Solve Any 2**)

1. Take two integer numbers as input and calculate their sum.
2. Calculate the area of a circle. Take radius as input.
3. Take some information about yourself (i.e. name, age, cgpa, department, section etc) as input and display them.
4. Read four integer values named A, B, C and D. Solve the following equation and print the result.
 - a) $(A * B - C * D)$
 - b) $2A - B + 3D$
 - c) $A^2 + B^2 - C^2 + D^2$
 - d) $A^3 + B - C^2$

Sample Input	Output Sample
5	Solution of Equation - 1 = -26
6	Solution of Equation - 2 = 28
7	Solution of Equation - 3 = 76
8	Solution of Equation - 4 = 67

Control Statement related Problems (**Solve Any 2**)

5. Find out the Maximum value from three integer numbers A, B and C.

Sample Input	Output Sample
A = 10 B = 5 C = 50	Maximum Value = 50

6. Create a **grading system** that will display the grade of your obtained mark.

Marks obtained out of 100	Grade	Grade point Equivalent	Remarks
80% and above	A+	4.00	Outstanding
75% to less than 80%	A	3.75	Excellent
70% to less than 75%	A-	3.50	Very Good
65% to less than 70%	B+	3.25	Good
60% to less than 65%	B	3.00	Satisfactory
55% to less than 60%	B-	2.75	Above Average
50% to less than 55%	C+	2.50	Average
45% to less than 50%	C	2.25	Below Average
40% to less than 45%	D	2.00	Pass
Less than 40%	F	0.00	Fail

7. Check whether the input is an **Alphabet** or a **digit** or a **Special Character**. If Alphabet then check whether it is **Vowel** or **Consonant**.

Sample Input	Output Sample
Test Case = 3 6 A @	6 is a Digit A is an Alphabet : It is an Vowel @ is a special character

LOOP related Problems (Any 1)

1. Write a Java program that takes a number as input and prints its **multiplication table** upto 10.

Sample Input	Output Sample
Input a number: 8	8 x 1 = 8 8 x 2 = 16 8 x 3 = 24 ... 8 x 10 = 80

2. Write a Java program to print all the numbers between 1 and 100 except 10, 20, 30, and 40 which are dividable by 3 and 6.
3. Write a Java Code to check whether an Input is a **Prime** number or not.

Sample Input	Output Sample
Test Case =2 5 6	5 is a prime number 6 is not a prim number

```
/*Prime or not*/
package labtask.pkg2;

import java.util.Scanner;

public class PrimeOrNot {
    public static void main(String[] args) {
        for(int j = 0; j <= 1; ++j) {
            System.out.println("Test case: " + (j + 1));
            Scanner input = new Scanner(System.in);
            int count = false;
            System.out.print("Enter a number: ");
            int a = input.nextInt();
            if (a != 0 && a != 1) {
                for(int i = 2; i < a; ++i) {
                    if (a % i == 0) {
                        System.out.println("Not a prime number");
                        count = true;
                        break;
                    }
                }
                if (!count) {
                    System.out.println(a + " is a prime number");
                }
            } else {
                System.out.println("Not a prime number");
            }
        }
    }
}
```

```
/*Number print*/
package labtask.pkg2;

public class NumberPrint {
    public static void main(String[] args) {
        for(int i = 0; i <= 100; ++i) {
            if (i != 10 && i != 20 && i != 30 && i != 40) {
                if (i % 3 != 0 && i % 6 != 0) {
                    System.out.println(i);
                } else {
                    System.out.print("");
                }
            } else {
                System.out.print("");
            }
        }
    }
}
```

```
---/*Multiplication table*/
package labtask.pkg2;

import java.util.Scanner;

public class MultiplicationTable {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the value: ");
        int m = input.nextInt();
        System.out.println("The multiplicaton table of" + m + "is ");

        for(int i = 1; i <= 10; ++i) {
            int result = m * i;
            System.out.println(m + " x " + i + " = " + result);
        }
    }
}
```

```
/*Information*/
package labtask.pkg2;

public class Information {
    public static void main(String[] args) {
        String name = "Golam kibria";
        int age = 20;
        double cgpa = 3.5D;
        String dept = "CSE";
        char c = 'B';
        System.out.println("Name is: " + name);
        System.out.println("Age is: " + age);
        System.out.println("Cgpa is: " + cgpa);
        System.out.println("Department is: " + dept);
        System.out.println("Section is: " + c);
    }
}

-----/*Find max*/
package labtask.pkg2;

import java.util.Scanner;

public class FindMax {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int a = input.nextInt();
        int b = input.nextInt();
        int c = input.nextInt();
        if (a > b && a > c) {
            System.out.println("Maximum is: " + a);
        } else if (b > a && b > c) {
            System.out.println("Maximum is: " + b);
        } else {
            System.out.println("Maximum is: " + c);
        }
    }
}
```

```

/*Equaiton*/
package labtask.pkg2;

import java.util.Scanner;

public class Equation {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int A = input.nextInt();
        int B = input.nextInt();
        int C = input.nextInt();
        int D = input.nextInt();
        System.out.println("Solution of equation-1: " + (A * B - C * D));
        System.out.println("Solution of equation-2: " + (2 * A - B + 3 * D));
        System.out.println("Solution of equation-3: " + (A * A + B * B - C * C
+ D * D));
        System.out.println("Solution of equation-4: " + (A * A * A + B - C *
C));
    }
}

-----
/*Char*/
package labtask.pkg2;

import java.util.Scanner;

public class Char {
    public static void main(String[] args) {
        for(int j = 0; j < 3; ++j) {
            System.out.println("Test case: " + (j + 1));
            Scanner input = new Scanner(System.in);
            System.out.print("Enter a character: ");
            char c = input.next().charAt(0);
            if ((c < 'A' || c > 'Z') && (c < 'a' || c > 'z')) {
                if (c >= '0' && c <= '9') {
                    System.out.println("Digit");
                } else {
                    System.out.println("Special Character");
                }
            } else if (c != 'a' && c != 'e' && c != 'i' && c != 'o' && c != 'u'
&& c != 'A' && c != 'E' && c != 'I' && c != 'O' && c != 'U') {
                System.out.println("Consonant");
            } else {
                System.out.println(c + " is an Alphabet: It is an Vowel");
            }
        }
    }
}
=====
```

```
/*Calculate sum*/
package labtask.pkg2;

import java.util.Scanner;

public class CalculateSum {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter two integer number: ");
        int a = input.nextInt();
        int b = input.nextInt();
        int sum = a + b;
        System.out.println("The sum is: " + sum);
    }
}

-----
/*Result*/
package labtask.pkg2;

import java.util.Scanner;

public class CalculateResult {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter your marks: ");
        int a = input.nextInt();
        if (a >= 80 && a <= 100) {
            System.out.println("A+ 4.0 Outstanding");
        } else if (a >= 75 && a <= 79) {
            System.out.println("A 3.75 Excellent");
        } else if (a >= 70 && a <= 74) {
            System.out.println("A- 3.5 Very good");
        } else if (a >= 65 && a <= 69) {
            System.out.println("B+ 3.25 Good");
        } else if (a >= 60 && a <= 64) {
            System.out.println("B 3.0 Satisfactory");
        } else if (a >= 55 && a <= 59) {
            System.out.println("B- 2.75 Above average");
        } else if (a >= 50 && a <= 54) {
            System.out.println("C+ 2.5 Average");
        } else if (a >= 45 && a <= 49) {
            System.out.println("C 2.25 Below average");
        } else if (a >= 40 && a <= 44) {
            System.out.println("D 2.00 Pass");
        } else {
            System.out.println("Fail");
        }
    }
}
```

```
-----  
/*Area of circle*/  
package labtask.pkg2;  
  
import java.util.Scanner;  
  
public class AreaOfCircle {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
        System.out.print("Enter the radius: ");  
        double r = input.nextDouble();  
        double area = 3.1416D * r * r;  
        System.out.println("The area of circle is: " + area);  
    }  
}
```

147. Creating a simple food order app.

```
package neso;

import java.util.Scanner;

public class ProjectOne {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Welcome to our food app");
        System.out.print("Please create your password: ");
        String p1 = input.nextLine();
        System.out.println("Your password is: " + p1);

        System.out.print("Please enter your password: ");
        String password = input.nextLine();
        while (!password.equals(p1)) {
            System.out.println("This is not the right password. Please try
again");
            password = input.nextLine();
        }
        System.out.println("Access granted");

        System.out.println("Are you hungry-\\"Yes\\" or \\"No\\\"");
        String a1 = input.nextLine();

        if (a1.equals("Yes")) {
            System.out.println("Choose what you want to eat between Pizza,
Burger and Fish");
            String a2 = input.nextLine();
            while (!a2.equals("Pizza") && !a2.equals("Burger") &&
!a2.equals("Fish")) {
                System.out.println("Sorry, we do not have it. What would you
like between Pizza, Burger and Fish");
                a2 = input.nextLine();
            }
            if (a2.equals("Pizza") || a2.equals("Burger") ||
a2.equals("Fish")) {
                System.out.println("Do you want something to drink? (Yes or
No)");
                String a3 = input.nextLine();
                while (!a3.equals("Yes") && !a3.equals("No")) {
                    System.out.println("Wrong command. Please try again");
                    a3 = input.nextLine();
                }
            }
        }
    }
}
```

```

        if (a3.equals("Yes")) {
            System.out.println("What would you like to drink? Coffee
or Juice");
            String a4 = input.nextLine();
            while (!a4.equals("Coffee") && !a4.equals("Juice")) {
                System.out.println("Sorry we do not have it. What
would you like between Coffee and Juice?");
                a4 = input.nextLine();
            }
            if (a4.equals("Coffee") || a4.equals("Juice")) {
                System.out.println("Great you have ordered " + a2 +
and " + a4);
            }
            System.out.println("Please confirm your password");
            password = input.nextLine();
            while (!password.equals(p1)) {
                System.out.println("This is not the right password.
Please try again");
                password = input.nextLine();
            }
            System.out.println("Thank you. Your order is confirmed and
this item is on its way");
        } else {
            System.out.println("Please confirm your password");
            password = input.nextLine();
            while (!password.equals(p1)) {
                System.out.println("This is not the right password.
Please try again");
                password = input.nextLine();
            }
            System.out.println("Thank you. You have ordered " + a2 +
and its confirmed and this item is on its way");
        }
    }
} else {
    System.out.println("OK. Have a good day.");
}
}
}

```

OUTPUT:

Welcome to our food app

Please create your password: 123

Your password is: 123

Please enter your password: 1

This is not the right password. Please **try** again

123

Access granted

Are you hungry- "Yes" or "No"

Yes

Choose what you want to eat between Pizza, Burger and Fish

Pizza

Do you want something to drink? (Yes or No)

Yes

What would you like to dring? Cofee or Juice

Juice

Great you have ordered Pizza and Juice

Please confirm your password

123

Thank you. Your order in confirm and **this** items are on it's way

BUILD SUCCESSFUL (total time: 29 seconds)

```
/*Polymorphism*/  
  
package udemy;  
  
class band {  
    //parent class  
    void perform() {  
        System.out.println("Performing");  
    }  
}  
  
class singer extends band {  
    //child class  
    void perform() {  
        System.out.println("Singing");  
    }  
}  
  
class drummer extends band {  
    void perform() {  
        System.out.println("Drumming");  
    }  
}  
  
public class Aone {  
  
    public static void main(String[] args) {  
        band obj1 = new band();  
        obj1.perform();  
  
        band obj2 = new singer();  
        obj2.perform();  
  
        band obj3 = new drummer();  
        obj3.perform();  
    }  
}
```

Output:
Performing
Singing
Drumming

```
/*Method overloading*/

package udemy;

public class Bone {

    int sum(int a, int b) {
        return a + b;
    }

    int sum(int a, int b, int c) {
        return a + b + c;
    }

    double sum(double a, double b) {
        return a + b;
    }

    public static void main(String[] args) {
        Bone obj1 = new Bone();
        int x1 = obj1.sum(1, 2);
        System.out.println(x1);

        Bone obj2 = new Bone();
        int x2 = obj2.sum(1, 2, 3);
        System.out.println(x2);

        Bone obj3 = new Bone();
        double x3 = obj3.sum(1.5, 2.5);
        System.out.println(x3);
    }
}
Output:
3
6
4.0
```

```
/*Method overriding*/

package udemy;

class car {

    void display() {
        System.out.println("Nice car");
    }
}

class whell extends car {

    void display() {
        System.out.println("Nice whell");
    }
}

public class Cone {

    public static void main(String[] args) {
        whell obj = new whell();
        obj.display();
    }
}
Output:
Nice whell
```

```
/*Super Keyword part-1*/  
  
package udemy;  
  
class A {  
    int a = 10;  
}  
  
class B extends A {  
    int a = 20;  
  
    void display() {  
        System.out.println("Value of a is: " + a);  
        System.out.println("Value of a is: " + super.a);  
    }  
}  
  
public class Done {  
    public static void main(String[] args) {  
        B obj = new B();  
        obj.display();  
    }  
}  
Output:  
Value of a is: 20  
Value of a is: 10
```

```
/*Super Keyword part-2*/  
  
package udemy;  
  
class G {  
  
    //we can also make a class by G(){}  
    void party() {  
        System.out.println("We are in party");  
    }  
}  
  
class K extends G {  
  
    //sudu G diye create korly only super(); hoby  
    void party() {  
        super.party();  
        System.out.println("Dhaka, Bangladesh");  
    }  
}  
  
public class Eone {  
  
    public static void main(String[] args) {  
        K obj = new K();  
        obj.party();  
    }  
}
```

Output:
We are in party
Dhaka, Bangladesh

Lab Task – 3

Array (**Solve Any 2**)

1. You have an Array of N numbers. Now write a code to **Sort** the elements in Ascending Order.

Input Samples	Output Samples
How many Numbers you want to insert? 5 Enter 5 array elements 10 220 303 140 50	10 50 140 220 303

2. You have an Array of N numbers. Now write a code to **Search** an element from an array **input** from user. [**Linear Search**]

Input Samples	Output Samples
Total Case = 2 How many Numbers you want to insert? 5 Enter 5 array elements 10 220 303 140 50 Case:1 Enter Data You want to search 220 Case:2 Enter Data You want to search 120	220 found at Index 1 120 not found in the Array

3. Suppose you have stored the CGPA of N number of students in an array. Now find the **smallest** and **largest** CGPA of the array.

Input Samples	Output Samples
How many Students' CGPA you want to insert? 5 Enter 5 array elements 3.8 3.9 3.3 3.75 2.8	Largest CGPA = 3.9 Smallest CGPA = 2.9

4. Suppose you have two Arrays. Now you need to **merge** those arrays in **one Single Array**.

Input Samples	Output Samples
N1 = 5 Enter 5 array elements 10 220 303 140 50 N2 = 3 Enter 5 array elements 400 500 600	10 220 303 140 50 400 500 600

Solving the Problems using Math class (**Solve Any 3**)

- Find **absolute**, **floor**, **ceil**, **round** and **square root** values of a number.
- Find the **maximum** and **minimum** values from **three numbers** using MATH Class.
- Generate **5 random** numbers between **0 and 200**.
- Calculate **2^0 to 2^n** Using Math Class. 'n' will be input from user.
- Calculate the **area** of a **circle** using Math.pow() and Math.PI methods

```
/*Math class-1*/
package mathclass;

public class MathClassOne {

    public static void main(String[] args) {

        int a = -7;
        int abs = Math.abs(a);
        System.out.println("Absolute value = " + abs);

        System.out.println("Floor = " + Math.floor(5.6));
        System.out.println("Ceil = " + Math.ceil(5.6));

        int round = Math.round(4.7f);
        System.out.println("Round number is = " + round);

        System.out.println("Square is = " + Math.sqrt(25));
    }
}
```

```
/*Math class-2*/
package mathclass;

import java.util.Scanner;

public class MathClassTwo {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        int a, b, c, x, y;
        System.out.print("Enter num1: ");
        a = input.nextInt();

        System.out.print("Enter num2: ");
        b = input.nextInt();

        System.out.print("Enter num3: ");
        c = input.nextInt();

        x = Math.max(a, b);
        x = Math.max(x, c);

        y = Math.min(a, b);
        y = Math.min(y, c);

        System.out.println("Maximum number is: " + x);
        System.out.println("Minimum number is: " + y);
    }
}

/*Math class-3*/
package mathclass;

import java.util.Random;

public class MathClassThree {

    public static void main(String[] args) {

        Random r = new Random();

        for (int i = 0; i < 5; i++) {
            System.out.println("Random number: " + r.nextInt(200));
        }
    }
}
```

```
/*Math class-4*/
package mathclass;

import java.util.Scanner;

public class MathClassFour {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        double sum = 0, n, i;
        System.out.println("Enter the ending power: ");
        n = input.nextDouble();
        System.out.println("2^0+2^1+2^2+-----+2^" + n);
        for (i = 0; i < n + 1; i++) {
            sum = sum + Math.pow(2, i);
        }
        System.out.println("The sum is: " + sum);
    }
}
```

```
/*Math class-5*/
package mathclass;

import java.util.Scanner;

public class MathClassFive {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        double r, area;
        System.out.print("Enter the radius: ");
        r = input.nextDouble();

        area = Math.PI * Math.pow(r, 2);

        System.out.println("The radius of circul is: " + area);
    }
}
```

```
/*Array one*/
package mathclass;

import java.util.Arrays;
import java.util.Scanner;

public class ArrayOne {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("How many numbers you wants to insert? ");
        int n = input.nextInt();
        System.out.printf("Enter %d array elements: ", n);
        int[] number = new int[n];
        for (int i = 0; i < n; i++) {
            number[i] = input.nextInt();
        }

        Arrays.sort(number);

        System.out.print("Ascending order: ");
        for (int i = 0; i < n; i++) {
            System.out.print(" " + number[i]);
        }
        System.out.println("");
    }
}
```

```
/*Array two*/
package mathclass;

import java.util.Scanner;

public class ArrayTwo {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Total case = ");
        int x;
        x = input.nextInt();

        System.out.print("How many numbers you wants to insert? ");
        int n;
        n = input.nextInt();
        int[] number1 = new int[n];

        System.out.println("Enter " + n + " array elements ");
        for (int i = 0; i < n; i++) {
            number1[i] = input.nextInt();
        }

        for (int k = 0; k < x; k++) {
            System.out.println("Case: " + (k + 1));
            int m;
            System.out.println("Enter value you want to search: ");
            m = input.nextInt();
            int i;

            for (i = 0; i < n; i++) {
                if (number1[i] == m) {
                    System.out.println(m + " found at index " + (i + 1));
                    break;
                }
            }
            if (i == n) {
                System.out.println(m + " not found in the array");
            }
        }
    }
}
```

```
/*Array three*/
package mathclass;

import java.util.Scanner;

public class ArrayThree {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        System.out.println("How many students CGPA you want to insert?");
        int n;
        n = input.nextInt();
        double[] number = new double[n];
        double sum = 0;

        System.out.println("Enter " + n + " array elements");
        for (int i = 0; i < n; i++) {
            number[i] = input.nextDouble();
        }

        double max = number[0];
        double min = number[0];
        for (int i = 1; i < number.length; i++) {
            if (max < number[i]) {
                max = number[i];
            }

            if (min > number[i]) {
                min = number[i];
            }
        }
        System.out.println("Largest CGPA = " + max);
        System.out.println("Smallest CGPA = " + min);
    }
}
```

```
/*Array four*/
package mathclass;

import java.util.Scanner;

public class ArrayFour {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        System.out.print("N1 = ");
        int n;
        n = input.nextInt();
        int[] number1 = new int[n];

        System.out.println("Enter " + n + " array elements ");
        for (int i = 0; i < n; i++) {
            number1[i] = input.nextInt();
        }

        int length1 = number1.length;

        System.out.print("N2 = ");
        int m;
        m = input.nextInt();
        int[] number2 = new int[m];
        System.out.println("Enter " + m + " array elements ");
        for (int i = 0; i < m; i++) {
            number2[i] = input.nextInt();
        }

        int length2 = number2.length;

        int length = length1+length2;

        int[] number3 = new int[length];
        for (int i = 0; i < length1; i++) {
            number3[i] = number1[i];
        }
        for (int i = 0; i < length2; i++) {
            number3[length1+i] = number2[i];
        }

        for (int i = 0; i < length; i++) {
            System.out.println(number3[i]);
        }

    }
}
```

Lab Task-4 solve

Lab Task - 4

Methods in JAVA (Solve Any 2)

1. Make a **simple calculator** using four methods (**addition, subtraction, multiplication** and **division**). Each Method will **receive the arguments** and will **return the result**. **[No-Static Method]**
2. Find the **area** of a **Circle**. Write a method which will **receive the radius as argument** and then will **return the area**. **[non-static method]**
3. Write a method named **maxValue()** which will find out the **maximum** value from **two integer** numbers. **[Static method with argument and return value]**
4. Write a method named **maxValue()** which will find out the **maximum element** from an **Array**. **[Static method with argument and return value]**

UML to JAVA Code (Solve Any 2)

1. Create the **Student** Class. Then create two **objects** of this class. Take User input for the instance variables.

Student	
-name:	String
-id:	int
-cgpa:	double
+main(String[]):void	

2. Create the **Student** Class. Then create **N Number objects** of this class. Take User input for the instance variables.

Student	
-name:	String
-id:	int
-cgpa:	double
<u>+main(String[]):void</u>	

3. Create the **Student** Class. Then create two **objects** of this class. Take User input for the instance variables.

Student	
-name:	String
-id:	int
-cgpa:	double
<u>+insertRecord (String , int , double) : void</u>	
<u>+displayRecord():void</u>	
<u>+main(String[]):void</u>	

4. Create the **Student** Class. Then create N number of **objects** of this class. Take User input for the number of objects and the instance variables [**Array of Objects**].

Student	
-name:	String
-id:	int
-cgpa:	double
<u>+insertRecord (String , String , double) : void</u>	
<u>+displayRecord():voidint</u>	
<u>+main(String[]):void</u>	

String Class in Java (**Solve Any 3**)

1. Concat (**Merge**) Three Strings together.

Input Samples	Output Samples
String - 1: Hello String - 2: My String - 3: Class	Hello My class

2. Input **Three Strings** and find out which one is **Longer**. You have to compares Three strings **lexicographically**

Input Samples	Output Samples
String - 1: Hello String - 2: Class String - 3: School	String 3 is greatest.

3. Input **Three Strings** and find out which two Strings are **Equal**.

Input Samples	Output Samples
String - 1: Hello String - 2: Class String - 3: Hello	String - 1 is Equals to String- 3

4. Store **Some Strings** and Display them in **Alphabetical Order**

Input Samples	Output Samples
Babul Kamal Alam Antara	Alam Antara Babul Kamal

- 5.** Input Two Strings and Convert them in both **Lower** case and **Upper** case

Input Samples	Output Samples
Babul Kamal	Lower Case: bakul kamal Upper Case: BAKUL KALA

Home Work

- 1.** **Unsolved** Problems from the **Lab Task - 4**.
- 2.** Suppose you are going to start your own company. You need to recruit some employees. Now, create a class named "**Employee**" to store the information of them. **Convert** the following **UML** into **Java code** and **store** information of three employees.

Employee
-name: String -id: String -age: int -salary: double
+setEmpNameId(String ,String) : void +setEmpAge(int) : void +setEmpSalary(double) : void +display_Info():void +main(String) : void

```
/*Method-1*/
package labfour;

import java.util.Scanner;

public class Test {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        double num1, num2;

        System.out.print("Enter first number: ");
        num1 = input.nextDouble();
        System.out.print("Enter second number: ");
        num2 = input.nextDouble();

        A ob = new A();
        ob.add(num1, num2);
        ob.sub(num1, num2);
        ob.mul(num1, num2);
        ob.division(num1, num2);
    }
}

package labfour;

public class A {

    void add(double x, double y) {
        System.out.println("Addition is: " + (x + y));
    }

    void sub(double x, double y) {
        System.out.println("Subtraction is: " + (x - y));
    }

    void mul(double x, double y) {
        System.out.println("Multiplication is: " + (x * y));
    }

    void division(double x, double y) {
        System.out.println("Division is: " + (x / y));
    }
}
-----
```

```
/*Method-2*/
package labfour;

import java.util.Scanner;

public class AreaOfCircul {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        double r, area;
        System.out.print("Enter the radius: ");
        r = input.nextDouble();

        Ct ob = new Ct();
        ob.radius(r);
    }
}

package labfour;

public class Ct {

    void radius(double x) {
        double result = Math.PI * x * x;
        System.out.printf("Area of a circle is: %.2f\n", result);
    }
}
-----
```

```
/*Method-3*/
package labfour;

import java.util.Scanner;

public class FindMax {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        int a, b, result;
        System.out.print("Enter the first number: ");
        a = input.nextInt();
        System.out.print("Enter the second number: ");
        b = input.nextInt();

        result = findMax(a, b);
        System.out.println("Maximum number is: " + result);
    }

    public static int findMax(int x, int y) {
        return +Math.max(x, y);
    }
}
```

```
/*Method-4*/
package labfour;

import java.util.Scanner;

public class MaxArray {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        System.out.print("How many numbers: ");
        int n;
        n = input.nextInt();
        double[] number = new double[n];

        System.out.println("Please enter " + n + " number = ");
        for (int i = 0; i < n; i++) {
            number[i] = input.nextDouble();
        }

        double max = maxValue(number);
        System.out.println("Maximum is: " + (int) max);

    }

    public static double maxValue(double[] number) {
        double max = number[0];
        double min = number[0];
        for (int i = 1; i < number.length; i++) {
            if (max < number[i]) {
                max = number[i];
            }
        }
        return max;
    }
}
```

```
/*UML-1*/
package labfour;

import java.util.Scanner;

public class Student {

    String name;
    int id;
    double cgpa;

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.println("Please enter s1 name, id and cgpa: ");
        Student s1 = new Student();
        s1.name = input.nextLine();
        s1.id = input.nextInt();
        s1.cgpa = input.nextDouble();

        System.out.println("Name: " + s1.name);
        System.out.println("ID: " + s1.id);
        System.out.println("CGPA: " + s1.cgpa);

        input.nextLine();

        System.out.println("\nPlease enter s2 name, id and cgpa: ");
        Student s2 = new Student();
        s2.name = input.nextLine();
        s2.id = input.nextInt();
        s2.cgpa = input.nextDouble();

        System.out.println("Name: " + s2.name);
        System.out.println("ID: " + s2.id);
        System.out.println("CGPA: " + s2.cgpa);

    }
}
```

```
/*UML-2*/
package labfour;

import java.util.Scanner;

public class Person2 {

    String name;
    int age;
    double cgpa;

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        Person2[] person = new Person2[5];
        System.out.print("How many students? ");
        int n = input.nextInt();

        for (int i = 0; i < n; i++) {
            person[i] = new Person2();
            System.out.print("Enter name: ");
            input.nextLine();
            person[i].name = input.nextLine();
            System.out.print("Enter age: ");
            person[i].age = input.nextInt();
            System.out.print("Enter CGPA: ");
            person[i].cgpa = input.nextDouble();
        }

        for (int i = 0; i < n; i++) {
            System.out.println("\nInformation of person: " + (i + 1));
            System.out.println("Name: " + person[i].name);
            System.out.println("Age: " + person[i].age);
            System.out.println("CGPA: " + person[i].cgpa);
        }
    }
}
```

```
/*UML-3*/
package labfour;

public class Students {

    String name;
    int id;
    double cgpa;

    void insertRecord(String name, int id, double cgpa) {
        this.name = name;
        this.id = id;
        this.cgpa = cgpa;
    }

    void displayRecord() {
        System.out.println("Name: " + name);
        System.out.println("ID: " + id);
        System.out.println("CGPA: " + cgpa);
    }
}
package labfour;

import java.util.Scanner;

public class TestJava {

    public static void main(String[] args) {

        Students s = new Students();
        Scanner input = new Scanner(System.in);

        System.out.print("Enter name: ");
        String name = input.nextLine();
        System.out.print("Enter ID: ");
        int code = input.nextInt();
        System.out.print("Enter CGPA: ");
        double cg = input.nextDouble();

        s.insertRecord(name, code, cg);
        s.displayRecord();
    }
}
-----
```

```
/*UML-4*/
package labfour;

public class Students {

    String name;
    int id;
    double cgpa;

    void insertRecord(String name, int id, double cgpa) {
        this.name = name;
        this.id = id;
        this.cgpa = cgpa;
    }

    void displayRecord() {
        System.out.println("Name: " + name);
        System.out.println("ID: " + id);
        System.out.println("CGPA: " + cgpa);
    }
}
package labfour;
import java.util.Scanner;

public class TestJava {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        int n = input.nextInt();
        Students[] s = new Students[n];

        for (int i = 0; i < n; i++) {
            s[i] = new Students();
            System.out.print("Enter name: ");
            input.nextLine();
            s[i].name = input.nextLine();
            System.out.print("Enter age: ");
            s[i].id = input.nextInt();
            System.out.print("Enter CGPA: ");
            s[i].cgpa = input.nextDouble();
        }

        for (int i = 0; i < n; i++) {
            s[i].insertRecord(s[i].name, s[i].id, s[i].cgpa);
            s[i].displayRecord();
        }
    }
}
```

```
/*UML-2 HOME WORK*/
package labfour;

public class Employee {

    String name;
    String id;
    int age;
    double salary;

    void setEmpNameId(String name, String id) {
        this.name = name;
        this.id = id;
    }

    void setEmpAge(int age) {
        this.age = age;
    }

    void setEmpSalary(double salary) {
        this.salary = salary;
    }

    void displayifo() {
        System.out.println("Name: " + name);
        System.out.println("ID: " + id);
        System.out.println("Age: " + age);
        System.out.println("Salary: " + salary);
    }
}
package labfour;

public class TestJava {

    public static void main(String[] args) {

        Employee e = new Employee();

        e.setEmpNameId("Rahim", "100");
        e.setEmpAge(20);
        e.setEmpSalary(1500.25);

        e.displayifo();
    }
}
```

```
/*String-1*/
package labfour;

import java.util.Scanner;

public class StringConcat {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("String-1: ");
        String s1 = input.nextLine();

        System.out.print("String-2: ");
        String s2 = input.nextLine();

        System.out.print("String-3: ");
        String s3 = input.nextLine();

        String s4 = s1.concat(" ").concat(s2).concat(" ").concat(s3);
        System.out.println(s4);
//        System.out.println(s1 + " " +s2 + " " +s3);
//        System.out.println(s1.concat(" ").concat(s2).concat(s3));
    }
}
```

```
/*string 2*/
package student;

import java.util.Scanner;

public class JavaPractice {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("String 1: ");
        String s1 = input.nextLine();

        System.out.print("String 2: ");
        String s2 = input.nextLine();

        System.out.print("String 3: ");
        String s3 = input.nextLine();

        int l1 = s1.length();
        int l2 = s2.length();
        int l3 = s3.length();

        if (l1 > l2 && l1 > l3) {
            System.out.println("String 1 is greatest");
        } else if (l2 > l1 && l2 > l3) {
            System.out.println("String 2 is greatest");
        } else {
            System.out.println("String 3 is greatest");
        }
    }
}
```

```
/*String-3*/
package student;

import java.util.Scanner;

public class JavaPractice {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("String 1: ");
        String s1 = input.nextLine();

        System.out.print("String 2: ");
        String s2 = input.nextLine();

        System.out.print("String 3: ");
        String s3 = input.nextLine();

        if (s1.equals(s2)) {
            System.out.println("String-1 is equal to String-2");
        }
        if (s1.equals(s3)) {
            System.out.println("String-1 is equal to String-3");
        }
        if (s2.equals(s3)) {
            System.out.println("String-2 is equal to String-3");
        }
    }
}
```

```
/*String-4*/
package student;

import java.util.Arrays;

public class JavaPractice {

    public static void main(String[] args) {
        String[] name = {"Kibria", "Saim", "Tausif", "Sazid", "Ariful"};

        Arrays.sort(name);
        for (int i = 0; i < 5; i++) {
            System.out.println(name[i]);
        }
    }
}
```

```
/*String-5*/
package student;

public class JavaPractice {

    public static void main(String[] args) {
        String[] name = {"Kibria", "Saim", "Tausif", "Sazid", "Ariful"};

        System.out.println("Upper Case: ");
        for (int i = 0; i < 5; i++) {
            System.out.println(name[i].toUpperCase());
        }

        System.out.println("\nLower Case: ");
        for (int i = 0; i < 5; i++) {
            System.out.println(name[i].toLowerCase());
        }
    }
}
```

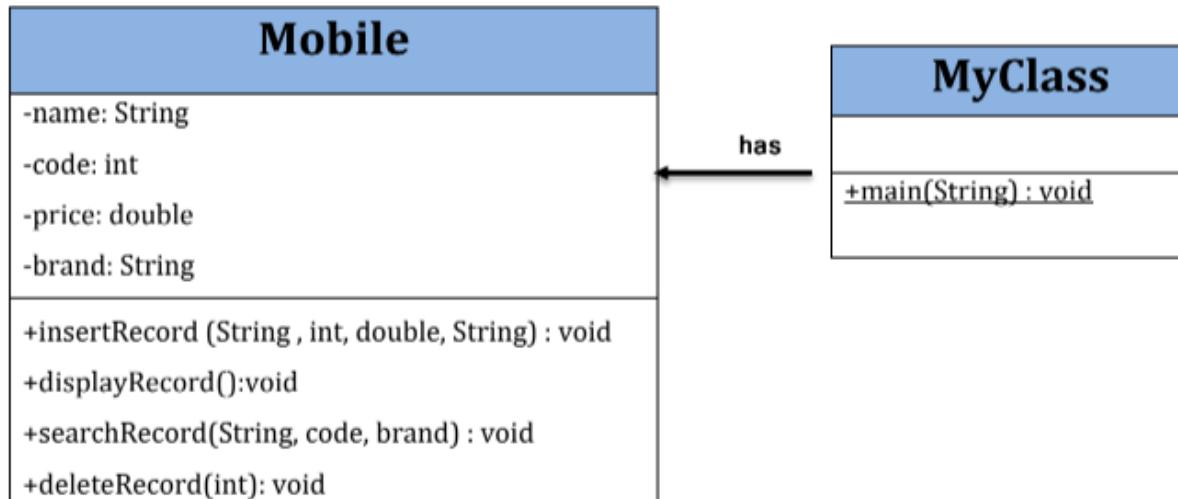
Lab Task - 5

UML to JAVA Code

Solve Any One (other one is Home work)

1. Convert the above UML into Java Code.

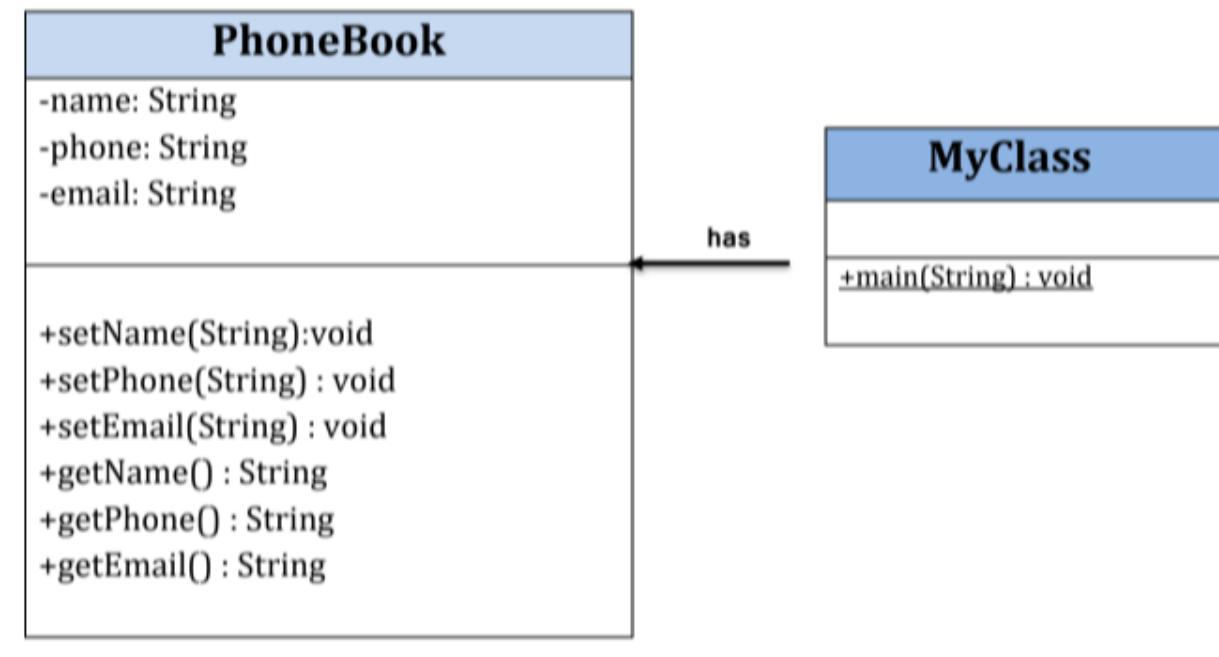
- **Store** details of Mobiles.
- **Display** all the Mobile List.
- **Search** for a friend by their **Name/Code/Brand**.
- You can **edit** or **delete** any their information.



- 2.** Suppose you are going to design a **PhoneBook** to store **contact details** of your **friends**.

You can -

- **Store Name, Phone Number and Email Address** of your Friends.
- **Display all** the Friend List.
- **Search** for a friend by their **Name/Phone/Email**.
- You can **edit** or **delete** any their information.



```
/*
-----1-----
-----
package student;

import java.util.ArrayList;
import java.util.Scanner;

public class Mobile {

    int code;
    double price;
    String name, brand;

    /*INSERT*/
    public void insertValue(int code, double price, String name, String
brand) {
        this.code = code;
        this.price = price;
        this.name = name;
        this.brand = brand;
    }

    /*DISPLAY*/
    public void displayValue() {
        System.out.println("Phone code: " + this.code);
        System.out.println("Phone price: " + this.price);
        System.out.println("Phone name: " + this.name);
        System.out.println("Phone brand: " + this.brand);

    }

    /*CREATE*/
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("How many times the loop will be executed? ");
        int n = input.nextInt();
        input.nextLine();

        Mobile[] m = new Mobile[n];
        for (int i = 0; i < n; i++) {
            m[i] = new Mobile();

            System.out.print("Enter the code: ");
            int phoneCode = input.nextInt();
            System.out.print("Enter the price: ");
            double phonePrice = input.nextDouble();
            input.nextLine();
            System.out.print("Enter the phone name: ");
            String phoneName = input.nextLine();
        }
    }
}
```

```

System.out.print("Enter the brand name: ");
String phoneBrand = input.nextLine();

//input.nextLine();
m[i].insertValue(phoneCode, phonePrice, phoneName, phoneBrand);
}

System.out.println("Information printing: ");
for (int i = 0; i < n; i++) {
    m[i].displayValue();
}

/*SEARCHING NAME*/
System.out.print("Enter the brand name you want to search: ");
String searchBrandName = input.nextLine();
int count = 0;
for (int i = 0; i < n; i++) {
    if (m[i].brand.equalsIgnoreCase(searchBrandName)) {
        count = 1;
    }
}
if (count == 1) {
    System.out.println("Brand name is found");
} else {
    System.out.println("Brand name is not found");
}

/*SEARCHING CODE*/
System.out.print("Enter the phone code you want to search: ");
int pCode = input.nextInt();
int c = 0;

for (int i = 0; i < n; i++) {
    if (m[i].code == pCode) {
        c = 1;
    }
}
if (c == 1) {
    System.out.println("Phone code is found");
} else {
    System.out.println("Phone code is not found");
}

/*REMOVING CODE AND NAME*/
ArrayList<Integer> r = new ArrayList<>();
ArrayList<String> s = new ArrayList<>();

System.out.print("Enter the code you want to remove? ");
int removeCode = input.nextInt();
r.add(removeCode);

```

```
//System.out.println(r);
r.remove(0);
System.out.println(r);
System.out.println("Code number Removed successfully");

input.nextLine();
System.out.print("Enter the brand name you want ot remove? ");
String removeBrand = input.nextLine();
s.add(removeBrand);
s.remove(0);
System.out.println(s);
System.out.println("Brand Name Removed Successfully");
}
}
```

```
2.
package labfive;

public class PhoneBook {

    String name, phone, email;

    public String getName() {
        return name;
    }

    public String getPhone() {
        return phone;
    }

    public String getEmail() {
        return email;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setPhone(String phone) {
        this.phone = phone;
    }

    public void setEmail(String email) {
        this.email = email;
    }

}

package labfive;

public class TestTwo {

    public static void main(String[] args) {

        PhoneBook p = new PhoneBook();
        p.setName("Golam kibria");
        p.setPhone("01790037447");
        p.setEmail("kibria14-522@diu.edu.bd");

        System.out.println(p.getName());
        System.out.println(p.getPhone());
        System.out.println(p.getEmail());
    }
}
```

156. Instance Initializer

```
package udemy;

class Count {

    int x;

    Count() {
        //super();
        //by default super keyword ta create hoye jay.
        System.out.println(x);
        System.out.println("Constructor");
    }

    {
        x = 10;
        System.out.println("Block is invoked");
    }
}

public class Fone {

    public static void main(String[] args) {
        Count obj = new Count();
    }
}
```

Output:

```
Block is invoked
10
Constructor
```

157. Aggregation in Java

```
package labfour;

public class Address {

    int street;
    String city, country;

    public Address(int street, String city, String country) {
        this.street = street;
        this.city = city;
        this.country = country;
    }
}

package labfour;

public class Employe{

    String name;
    int id;
    Address add;

    public Employe (String name, int id, Address add){
        this.name = name;
        this.id = id;
        this.add = add;
    }

    public static void main(String[] args) {
        Address ob1 = new Address(40, "Dhaka", "Bangladesh");
        Employe ob2 = new Employe("Kibria", 522, ob1);

        System.out.println("Employee: \nName: "+ob2.name);
        System.out.println("ID: "+ob2.id);

        System.out.println("\nAddress: \nStreet: "+ob2.add.street);
        System.out.println("City: "+ob2.add.city);
        System.out.println("Country: "+ob2.add.country);
    }
}
```

```
/*Alternative way*/
package labfour;

public class Address {

    int street;
    String city, country;

    public Address(int street, String city, String country) {
        this.street = street;
        this.city = city;
        this.country = country;
    }
}

package labfour;

public class Employe{

    String name;
    int id;
    Address add;

    public Employe (String name, int id, Address add){
        this.name = name;
        this.id = id;
        this.add = add;
    }

    public void display(){
        System.out.println("Employee: \nName: "+name);
        System.out.println("ID: "+id);

        System.out.println("\nAddress: \nStreet: "+add.street);
        System.out.println("City: "+add.city);
        System.out.println("Country: "+add.country);
    }
    public static void main(String[] args) {
        Address ob1 = new Address(40, "Dhaka", "Bangladesh");
        Employe ob2 = new Employe("Kibria", 522, ob1);

        ob2.display();
    }
}
```

158. Setter and Getter method using Constructor

```
package labfour;

public class Person {

    String name;
    int age;

    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }
}

package labfour;

public class MyClass {

    public static void main(String[] args) {

        Person ob = new Person("Kibria", 20);
        System.out.println("Name: "+ob.getName());
        System.out.println("Age: "+ob.getAge());

        ob.setName("Ezaz");
        ob.setAge(21);

        System.out.println("After update information: ");
        System.out.println("Name: "+ob.getName());
        System.out.println("Age: "+ob.getAge());
    }
}
```

Output:

Name: Kibria

Age: 20

After update information:

Name: Ezaz

Age: 21

