

OOP LAB # 03

Objective: To get familiarity with the operators and type casting.

Exercise #01: Write a program which calculates the Body Mass Index (BMI) of the user input values.

Also typecast the result if needed. If your BMI is less than 18.5, it falls within the Under Weight range. If your BMI is 18.5 to 24.9, it falls within the normal or Healthy Weight range. If your BMI is 25.0 to 29.9, it falls within the Over Weight range.

Formula: weight (kg) / [height (m)]² weight in kilograms divided by height in meters squared.

Example: Weight = 68 kg, Height = 165 cm (1.65 m)

Calculation: $68 \div (1.65 \times 1.65) = 24.98$.

Source code:

```
package bmi;
import java.util.Scanner;
public class BMI {
    public static void main(String[] args) {
        double Hg,Wg,bmi;
        Scanner sc=new Scanner(System.in);
        System.out.print("Plz enter your weight in kg:");
        Wg=sc.nextDouble();
        System.out.print("Plz enter your height in meters:");
        Hg=sc.nextDouble();
        bmi=Wg/(Hg*Hg);
        System.out.println("Size of bmi in bit is:"+Double.SIZE);
        System.out.println("Size of bmi in bytes is:"+Double.SIZE/8);
        float bmi_f=(float)bmi;
        System.out.println("Size of bmi in bit after typecasting is:"+Float.SIZE);
        System.out.println("Size of bmi in bytes after typecasting is:"+Float.SIZE/8);
        System.out.printf("Your BMI in bit is:%.2f\n",bmi_f);
        if(bmi<18.5){
            System.out.println("You are in under weight range!");
        }
        else if(bmi>18.5&&bmi<24.9){
            System.out.println("You are in Healthy weight range!");
        }
        else if(bmi>24.9&&bmi<=29.9){
            System.out.println("You are in Over weight range!"); } } }
```

Output:

```
Plz enter your weight in kg:68
Plz enter your height in meters:1.65
Size of bmi in bit is:64
Size of bmi in bytes is:8
Size of bmi in bit after typecasting is:32
Size of bmi in bytes after typecasting is:4
Your BMI in bit is:24.98
You are in Over weight range!
```

Exercise #02: Write a program which shows the implementation of increment and decrement operator.

Source code:

```
package ioincdec;
import java.util.Scanner;
public class IoIncDec {
    public static void main(String[] args) {
        int a;
        Scanner sc=new Scanner(System.in);
        System.out.print("Plz enter any number:");
        a=sc.nextInt();
        System.out.println("As it is post increment so Ans will be set to a:"+(a++));
        System.out.println("To show post increment we will now just print a"+a);
        System.out.println("As it is post decrement so Ans will be set to a:"+(a--));
        System.out.println("To show post decrement we will now just print a"+a);
        System.out.println("As it is pre increment so Ans will be set to a=a+1:"+(++a));
        System.out.println("As it is pre decrement so Ans will be set to a=a-1:"+(--a));
    }
}
```

Output:

```
Plz enter any number:5
As it is post increment so Ans will be set to a:5
To shw post increment we will now just print a6
As it is post decrement so Ans will be set to a:6
To show post decrement we will now just print a5
As it is pre increment so Ans will be set to a=a+1:6
As it is pre decrement so Ans will be set to a=a-1:5
```

Exercise #03: Write a program which shows the implementation of any three bit wise operators

Source code:

```
package iobo;
public class IoBO {
    public static void main(String[] args) {
        String binary[]={ "0000", "0001", "0010", "0100", "0101", "0110",
        "0111", "1000", "1001", "1010", "1011", "1011", "1100", "1101", "1110", "1111"};
        int a=1,b=2;
        int c=a|b,d=a&b,e=a^b;
        System.out.println(binary[a]);
        System.out.println(binary[b]);
        //bitwise OR working
        System.out.println("0001+0010="+binary[c]);
        //bitwise And working.it will multiply every digit and gives output
        System.out.println("0001*0010="+binary[d]);
        //bitwise XOR working
        System.out.println("0001^0010=10"+binary[e]);
    }
}
```

Output:

```
0001
0010
0001+0010=0100
0001*0010=0000
0001^0010=100100
```

Exercise #04: Write a program which shows the implementation of widening type casting and narrowing type casting.

Source code:

```
package typecasting;
public class Typecasting {
    public static void main(String[] args) {
        int a;
        long b=4;
        double c;
        float d=5;
        a=(int)b; //Manual casting: long to Int
        c=d; //automatic casting:double to float
    }
}
```

Exercise #05: Write a program that generates numbers randomly between 1 to 25 and convert numbers to their corresponding characters.

Source code:

```
package randomgenerator;
import java.util.Random;
public class Randomgenerator {
    public static void main(String[] args) {
        Random RNum = new Random();
        int Random_num=0+RNum.nextInt(25);
        System.out.println("Your random number is:"+Random_num);
        c=(char)Random_num;
        System.out.printf("Your random character is:%c\n",c);
    }
}
```

Output:

Your random number is:18

Your random character is: