



Topic $\Rightarrow$ User Defined Methods

(Constructors & Call by value & reference)

Lecture # 12

Data Type	Default Value
byte	0
Short	0
int	0
long	0L
float	0.0f
double	0.0
char	'\u0000'
String	null
boolean	false

h/w 8



Design a class FruitJuice as follows :

Data members / Instance Variable

int product_code --> To store product code number

String flavour --> To store flavour of Juice

String pack_type --> Type of bottle (Eg: Tetrapack, PETB, etc.)

int pack_size --> (500ml, 1000ml, etc.)

int product_price --> (Rs.10, Rs.20, Rs.95, etc.)

Member methods :

FruitJuice () : Default constructor to initialise int data members by 0 and String by null.

void input () : Input and store the details

void discount() : to reduce the product price by Rs.10

Also write main method to create object and call above function.

Ans in next slide

```
import java.util.Scanner;
class FruitJuice
{
    int product_code;
    String flavour;
    String pack_type;
    int pack_size;
    int product_price;

    FruitJuice()
    {
        product_code = 0;
        flavour = null;
        pack_type = null;
        pack_size = 0;
        product_price = 0;
    }
}
```

```
void input()
{
    Scanner ob = new Scanner(System.in);
    System.out.println("Enter Product code, flavour, pack type, pack size, price");
    product_code = ob.nextInt();
    ob.nextLine(); ← To remove the new line character
    flavour = ob.nextLine();
    pack_type = ob.nextLine();
    pack_size = ob.nextInt();
    product_price = ob.nextInt();
}

void discount()
{
    product_price = product_price-10;
}

void display()
{
    System.out.println("Product code = "+ product_code);
    System.out.println("Flavour = "+flavour);
    System.out.println("Pack type = "+ pack_type);
    System.out.println("Pack size = "+ pack_size);
    System.out.println("Product price after discount = "+ product_price);
}

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

Call by value

1. Functions are called by passing primitive data. (eg:- byte, int, double, etc.)
2. Formal parameters are copies of actual parameters.
3. Any change in the formal parameter will not affect actual parameter.

Call by reference

1. Functions are called by passing reference data. (eg:- objects)
2. Formal parameters are pointers of actual parameter
3. Any change in the formal parameter will affect the actual parameter.



Call by Value example :-



```
class Fun1
{
    int a=20;
    void change(int n)
    {
        n=n+10;
    }
    public void display ()
    {
        Fun1 ob1 = new Fun1();
        System.out.println("Value of a before change = "+a); //20
        change(a);
        System.out.println("Value of a after change = "+a); // 20
    }
    public static void main(String args[])
    {
        Fun1 ob = new Fun1();
        ob.display();
    }
}
```

Output

BlueJ: Terminal Window - FUN

Options

```
Value of a before change = 20
Value of a after change = 20
```


Call by reference example:-



```
class Fun
{
    int a=20;
    void change(Fun obj)
    {
        obj.a = obj.a+10;
    }
    public void display ()
    {
        Fun ob1 = new Fun();
        System.out.println("Value of a before change = "+ob1.a); //20
        change(ob1);
        System.out.println("Value of a after change = "+ob1.a); // 30
    }
    public static void main(String args[])
    {
        Fun ob = new Fun();
        ob.display();
    }
}
```

Output

BlueJ: Terminal Window - FUN

Options

Value of a before change = 20
Value of a after change = 30

Homework $\Rightarrow$ * Revise Call by
value & call by reference

* Revise class IX syllabus
for backup classes.

* Revise well for the test

Thank you