

OBJECT PASSING TO A FUNCTION IN JAVA

In Java as we do programs using the OOPS concept and class and its objects are the basic elements of programming in OOPS, we have the advantage of using the object as a user-defined data type and thus we can use the concept of passing objects to functions in the program.

We generally use this concept in programs where class is used as a composite data type and we need to work on the composite data as a single unit. Let us go through some examples:

- ❖ **Date (day/month/year)**
- ❖ **Time (hour/minute/second)**
- ❖ **Angle (degree/minute)**
- ❖ **Age (year/month/day)**
- ❖ **Height (feet/inch)**
- ❖ **Distance (km/m)**

An object can be passed to a function in two ways:

- *As a parameter* to the called function – in this case, the receiving function will have one or more objects of a particular class as its parameter and work upon those sets of data. The caller function will pass the objects of the class in the function call statement.
- *As a return type* to the caller function – in this case, the called function will return an object of a particular class as a return value to the caller function and the caller function needs to have an appropriate object at the receiving end

Note : In object passing, the call by reference or call by address concept is implemented and thus the change in formal parameters will reflect back in actual parameters too (if that occurred in the called function).

Let us see some real-life programs where we need this concept.

Suppose you have to find the age as on the current date in the year/month/day form, for filling up one application form, then what you will do? You have to subtract your date of birth from your current date.

For example on 01-01-2020 the age of a person whose date of birth is 21-11-1968 will be this one –

Curr Date : 2020 01 01 (Object A)

DOB : 1968 11 21 (Object B)

	10 (31-21)	
(30 added to CurrDate's day & 1 subtracted from month)01 (12-11)		
(12 added to CurrDate's month & 1 subtracted from yr)0051 (2019-1968)		
	51 01 10 (Object C)	

Thus the answer is: 51 years 1 month and 10 days

class AgeCalc

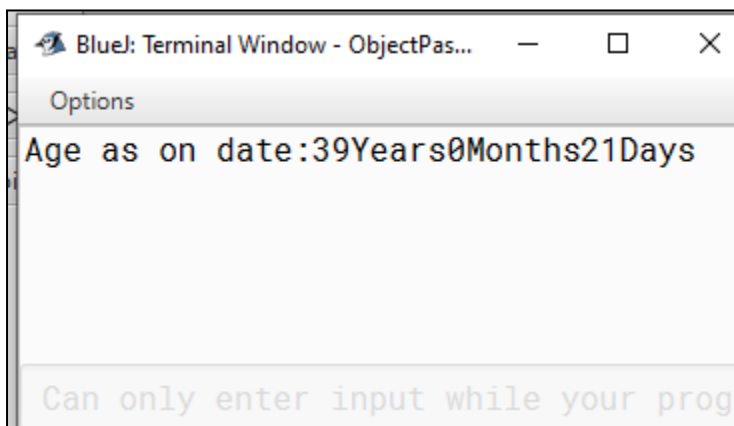
```

{
    int day, mnt, yr; //DATA MEMBERS
    public AgeCalc() //DEFAULT constructor
    {
        day=mnt=yr=0;
    }
    public AgeCalc(int a, int b, int c) //parameterised constr.
    {
        day=a;
        mnt=b;
        yr=c;
    }
    public void ageCalculator(AgeCalc DOB, AgeCalc COD)
    { // two objects, DOB and COD, represent date of birth and current date res.
        if(COD.yr<DOB.yr)
        {
            System.out.println("Invalid date");
            return;
        }
        if(DOB.day>COD.day)
        {
            COD.day+=30;
            COD.mnt--;
        }
    }
}

```

```
if(DOB.mnt>COD.mnt)
{
    COD.mnt+=12;
    COD.yr--;
}
this.day = COD.day - DOB.day;
this.mnt = COD.mnt - DOB.mnt;
this.yr = COD.yr - DOB.yr;
}
public void display()
{
    System.out.println("Age as on date:"+yr+"Years"+mnt+"Months"+day+"Days");
}
public static void main(String ar[])
{
    AgeCalc D1=new AgeCalc(10,12,1980); //Date of Birth
    AgeCalc D2=new AgeCalc(1,1,2020); //current date
    AgeCalc age=new AgeCalc(0,0,0);
    age.ageCalculator(D1, D2);
    age.display();
}
}
```

Output:



The screenshot shows a terminal window titled "BlueJ: Terminal Window - ObjectPas...". The window contains the following text:

```
Options
Age as on date:39Years0Months21Days
Can only enter input while your prog
```

Question 2.

Program to subtract one distance from another distance.

The class structure is given below:

Class name: Distance

Data members:

- int feet
- int inch

Member functions:

- | | |
|---|---|
| <ul style="list-style-type: none"> Distance(int, double) void display() Distance subtract(Distance) | <p>parameterized constructor
to display the total distance covered
to subtract the distance of the object passed as a parameter
from the current object and return the total distance
covered</p> |
|---|---|

Specify the class Distance giving the details of the constructor and all the functions. Write the main() method and create objects and call the functions required.

```
class Distance
{
    int feet;
    double inch;
    Distance(int ft, double in)
    {
        feet=ft;
        inch=in;
    }
    void displayDiff()
    {
        System.out.println("Difference in the distance: "+feet+" ft & "+ inch+ " inch");
    }
    Distance subtract(Distance D) // function will subtract the distance of D from the
    {
        // distance of the current object (this)
        Distance temp=new Distance(0,0); //temporary object
        if(this.inch < D.inch)
        {
            this.inch+=12;
            this.feet--;
        }
        temp.feet=this.feet-D.feet;
        temp.inch=this.inch-D.inch;
        return temp; //returning the temporary object which is holding the result
    }
    public static void main()
    {
        Distance A=new Distance(10,6); //first distance with 10 ft and 6 inches
        Distance B=new Distance(5,4); //second distance with 5 ft and 4 inches
        Distance Res=new Distance(0,0);
        Res=A.subtract(B); // actual subtraction of two distances is taking place,
        // A is the current object w.r.t. the function call here
        Res.displayDiff();
    }
}
```

Question 3.

A point on a 2D plane contains two coordinates – x and y. From a given two points P and Q, there exists a mid-point M between them whose coordinates are $\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2}$. Also the distance between P and Q can be calculated using the formula $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

```
class Point
{
    int x, y;

    Point(int i,int j)
    {
        x=i;
        y=j;
    }

    void display()
    {
        System.out.println("X "+x+", Y "+y);
    }

    void midPoint(Point P, Point Q)
    {
        x=(P.x+Q.x)/2;
        y=(P.y+Q.y)/2;
    }

    void findDist(Point P, Point Q)
    {
        int x1=Q.x-P.x;
        int y1=Q.y-P.y;
        double d=Math.sqrt((x1*x1)+(y1*y1));
        System.out.println(d);
    }

    public static void main()
    {
        Point P=new Point(4,6);
        Point Q=new Point(10,12);
        Point R=new Point(0,0);
        R.midPoint(P,Q);
        R.display();
        R.findDist(P,Q);
    }
}
```

~:Thank you :~