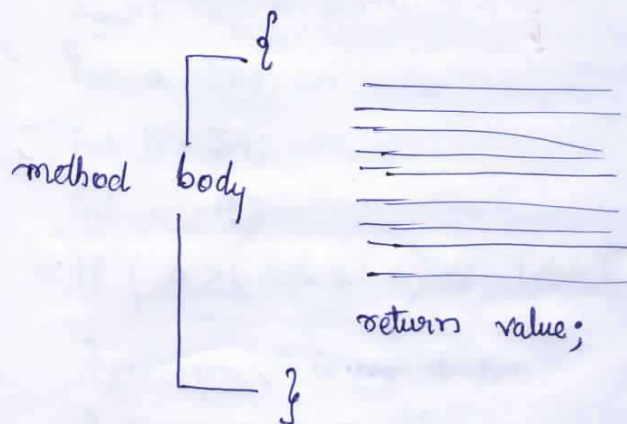


16-01-2013 WEDNESDAY

METHODS

SYNTAX:

modifiers returnType methodName (args0, args1, args2) data passed



// Mandatory Statement

Example:

P16

class Demo1

```

{
    psvm (String[] args)
    {
        S.o.p ("Program starts...");
        Sample ();
        S.o.p ("Program ends...");
    }
}

```

// invoking method (or) Calling method

Static int Sample();

```

{
    S.o.p ("running Sample method");
    return 10;
}
}

```

%p - Program starts...
running Sample method
Program ends...

*

```

6. int ret = sample();
7. S.o.p ("value of ret is " + ret);

```

%p = Program starts...
running Sample method
value of ret is 10
Program ends...

Example 2 :

P17

```
class Demo2
{
    psvm (String[] args)
    {
        Sop ("Program starts...");
        Sample();
        Sop ("Program ends...");
    }
    static void Sample()
    {
        Sop, out, println ("running sample method");
        // compiler writes return statement here
    }
}
```

O/p: Program starts...
running sample method
Program ends...

I.Q

Is return statement mandatory in a method in java?

Ans: Yes, but in case of a void method [compiler writes return statement] No.

Example 3 :

P18

```
class Demo3
{
    psvm (String[] args)
    {
        Sop ("Program starts...");
        int res = Sample (12, 34);
        Sop ("result is " + res);
        Sop ("*****");
        int res = Sample (23, 64);
        Sop ("result is " + res);
        Sop ("Program ends...");
    }
}
```

```
Static int sample(int a, int b)
```

```
{
```

```
    Sop("running sample method");
```

```
    Sop("value of a is "+a);
```

```
    Sop("value of b is "+b);
```

```
    int sum = a+b;
```

```
    return sum;
```

```
}
```

```
}
```

%P:

Program starts...

running sample method

value of a is 12;

value of b is 34

value of res is 46

running sample method

value of a is 23

value of b is 64

value of res is 87

Program ends...

NOTES

22*) While developing programs the repetitive tasks/operation are developed as method.

23*) method is a block when invoked executes all the statements of method body.

24*) The Syntax to develop a method in java is

modifiers return type methodName(arg0, arg1, arg2)

```
{
```

```
    return value;
```

```
}
```


- 25*) In every method return statement is mandatory.
- 26*) A return statement indicates the type of value the method has to return.
return value should match return type.
- 27*) Args ~~are~~ ~~used~~ ~~to~~ variable used to store data passed for method.
- 28*) A void return type indicates that method returns nothing.
- 29*) If a method return type is declared as void it is not mandatory to write return statement, in such case compiler writes return statement at the time of compilation.

Assign

1. S.I , Degree to F , Area of Circle using method.

P19

```
class AssignCombine
{
    Psum(String[] args)
    {
        Sop("Program starts...");

double res = SimpleInterest(10000, 0.15, 2);
        Sop("Simple Interest is " + res);
        Sop("*****");
        double res = degreeToFahrenheit(28);
        Sop("Fahrenheit is " + res);
        Sop("*****");

        double area = areaOfCircle(20);
        Sop("Area of Circle is " + area);
        Sop("*****");
        Sop("Program ends...");
    }
}
```

```

static double simpleInterest (double a, double rate, int time)
{
    System.out.println("Running simpleInterest method");
    double interest = a * rate * time;
    Sout("The amount " + a + " at the rate of " + rate + " for " + time + " years,");
    return interest;
}

```

```

static double degreeToFahrenheit (double degreeCelsius)
{
    Sout("Running Degree Celsius to Fahrenheit");
    double res = (degreeCelsius * 1.8) + 32;
    S.out("degreeCelsius + " Degree Celsius is equal to");
    return res;
}

```

```

static double areaOfCircle (double radius)
{
    S.out("Running area of Circle method");
    final double PI = 3.14;
    double area = PI * radius * radius;
    S.out("For radius " + radius);
    return area;
}
}

```