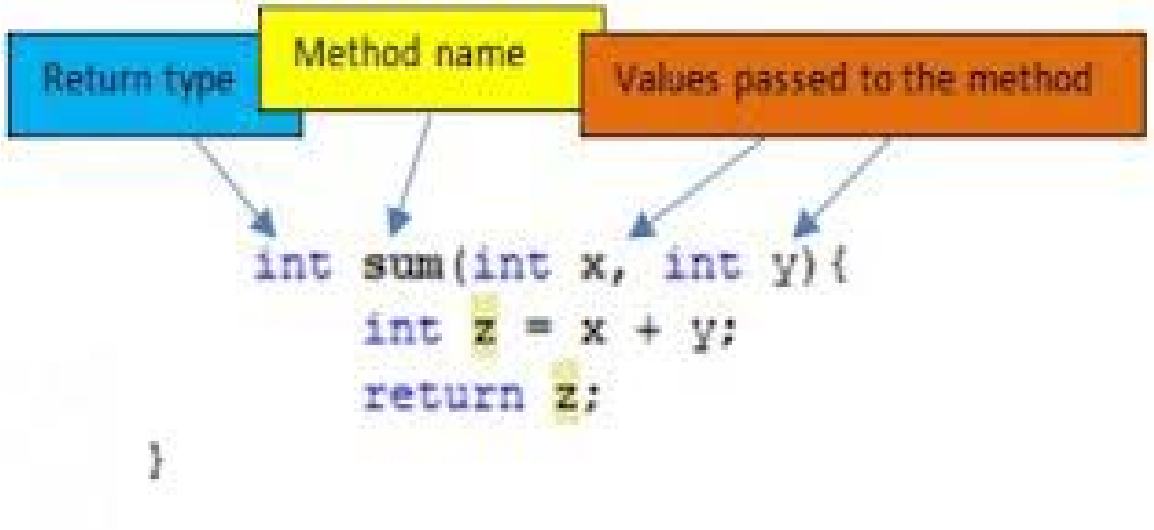
What are methods

- Methods are block of statements which

are used to do a specific task.

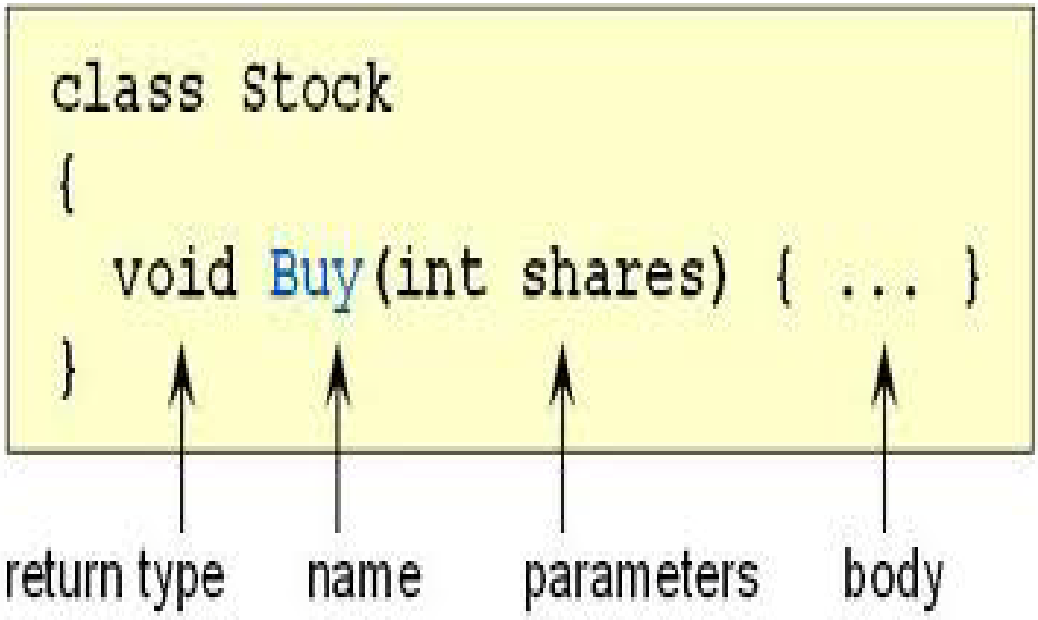
- Methods are generally used to divide a large code into manageable chunks of codes.



Method Declaration

1. Access/NonAccess Modifier
2. returnType
3. MethodName (Parameter List)
4. {
5. //method body or statement block
6. }

```
public void go(int x)
{
System.out.println(x);
}
```



Method Return Type

- It May be possible that a method returns a value or doesn't returns any value.

Return type of a method is nothing but the data type of the value returned by the method.

e.g.

- If method returns an integer value then it's return type will be 'int', for float 'float', for character 'char', for boolean 'boolean' and so on.
- If a method doesn't returns any value then it's return type will be "**void".**

- If a method returns a value then the last statement should be a **"return statement"**
- Any code after return statement is not reachable

Parameter List

- Method Parameters are the variables which are declared in the declaration of method.

e.g.

public int go(int x) {}

- The method parameters can be nothing, one or more than one. For multiple parameters they are separated by comma.

e.g.

Public int go (int x, float y, String s, char c){ }

Method Body

- Body of a method starts from opening curly braces ‘{‘ and ends at closing curly braces ‘}’.
- All statements of methods goes within the pair of curly braces.
- Execution of method stops on either ‘return’ statement or closing curly braces ‘}’.

e.g.

```
public int go(int x, int y)
{
Int  sum = x + y;
Return sum;
}
```

But how to call a method???????

Instance variable Vs Local variable

Instance Variables

1. Instance variables are those variables which are declared within a **class**.
2. These are known as ‘instance variables’ because each instance of the class (object) have it’s own copy of instance variable.

Local Variable

1. Local variable are those variables which are declared within a **method.**
2. These are known as ‘Local variable’ because they formed within method and dies at the end of the method

Accessing Object's methods and instance variables

- The methods and instance variables of any object can be accessed by the help of dot operator (.) on reference variable.
- referenceVariable.instanceVariable/method

e.g.

For class Animal having eat() method :

```
Animal a = new Animal();
a.eat();
```

Method Arguments and method calling

- Arguments are the values which are passed during calling of a method. These arguments are stored in the parameters of the method.
- Number of arguments must be equal to the number of parameters
- Calling a void method:

```
a.eat();
```

- Calling a single argument method:

```
a.eat(45);
```

- Calling a multiple argument method:

```
a.eat(343, 43, 'c', "wsfsf");
```

Predefined Methods

- Predefined methods are those methods which are already defined in java and ready to use.
- We can call these methods and can use in our code directly.

e.g.

Math.random();

- Math.random() is a method defined in Math class which is used to generate a random number between o.o to .9 in double type
- Calling Math.random() :

```
Int x = (int) (Math.random() * 4);

//It will generate the number from 0 to 3
```

Method Calling Methods

- The method can call other methods in chain-like pattern.

e.g.

```
Public class Test
{
public void firstMethod()
{
Test2 t = new Test2();
t.secondMethod(); // calling other method
}//end of method
}//end of class
```

IF, IF-Else, IF-Else-IF statements

- IF statement:

This is a decision making statement, which will execute it's codes ***if and only if*** the condition is true.

```
If(condition)
statement; // codes
```

- If-else statement:

In this the '***else***' codes will be execute ***if and only if*** the condition is false.

```
If(condition)
statement; //codes
else
statement; // codes
```

- If-else-if :

This is combination of if-else statements.

```
If(condition)
Statements;
Else if (condition)
Statements
Else if (condition)
Statements;
Else
Statements;
```

Guess-game

```
import java.util.*;
public class Test2 {
public static void main(String[] args)
{
Test2 t = new Test2();
Scanner scan = new Scanner(System.in); //ignore
System.out.println("enter a value");
int input = scan.nextInt(); //ignore
scan.close(); //ignore
int randomNum = (int)(Math.random()* 10);
if(input == randomNum)
System.out.println("well done smarty");
else
t.looser();
}
}
```

Guess-game continues

```
public void looser()
{
int looserNum = (int)(Math.random()*7);
if (looserNum == 0)
  System.out.println("u r so stupid");
else if (looserNum == 1)
  System.out.println("u r rediculus");
else if (looserNum == 2)
  System.out.println("why r u on earth!!! go to venus");
else if (looserNum == 3)
  System.out.println("u should be hanged out");
else if (looserNum == 4)
System.out.println(" don't dare to play it again");
else
System.out.println(" leave java now!!!");
}
```