**User Defined Methods in Java**  
   
i) Introduction to Java Methods  
  
ii) Types of Methods  
  
iii) User defined Methods  
-----------------------------------  
**i) Introduction to Java Methods**  
  
1) What is Method?  
   
> A Java method is a set of statements that are grouped together to perform an operation.  
  
> Methods are also known as Functions  
  
> In Structured programming (ex: C Language) we use Functions (Built in and User defined)  
  
> In Object Oriented Programming we use Methods (Built in and User defined)  
  
2) When we choose Methods?  
   
Whenever we want to perform any operation multiple times then choose methods.  
  
3) Advantages of Methods  
   
Code Reusability and Reduce the project code size.  
---------------------------------  
ii) Types of Methods  
   
Basically we have 2 types of Methods  
  
    1) Built in Methods  
  
    2) User defined Methods  
-----------------------------  
1) Built in Methods  
   
Java has a library of classes and methods organized in packages.  
  
Ex:  
  
import java.io.Console;  
  
import java.io.\*;  
  
> In order to use built in methods we need to import packages or classes.  
  
> java.lang package is automatically imported in every Java program.  
  
> Using import keyword we can import packages/classes.  
-----------------------------------------  
Categories of Built in Methods.  
   
    1) String Methods  
  
    2) Array Methods  
  
    3) Number Methods  
  
    4) Character Methods  
    Etc....  
---------------------------------  
**iii) User defined Methods**  
  
**Two types of User defined Methods**  
   
1) Method without returning any value  
        a) Calling Method by invoking Object  
        b) Calling Method without invoking Object  
  
2) Method with returning a value.  
-------------------------------  
        a) Calling Method by invoking Object  
        b) Calling Method without invoking Object  
---------------------------------------------------  
1) Writing Methods (With returning a value)  
  
a) Syntax for creating a method (calling the method by invoking object)  
   
Write Method  
//Before main Method  
accessModifier returnType methodName(Parameters){  
Statements  
---------  
---------  
------  
}  
----------------------  
//After main method  
Call Method  
//Create Object  
ClassName objectName = new ClassName();  
  
//Call Method y invoking object  
dataType variableName = object.method(values..);  
Or  
System.out.println(object.method());  
-----------------------------------  
Example:  
   
package xyza;  
  
public class JavaMethods {  
  
//User defined Method  
public int multiply(int a, int b, int c){  
int result = a \* b \* c;  
return result;  
}  
  
public static void main (String [] args){  
  
//Create Object  
JavaMethods abc = new JavaMethods();  
  
//Call Method  
int x = abc.multiply(10, 25, 35);  
System.out.println(x);  
      
System.out.println(abc.multiply(10, 25, 35));  
}  
}  
------------------------------------------------  
1) Writing Methods (With returning a value)  
  
b) Syntax for creating a method (calling the method without object)  
   
accessModifier nonAccessModifier returnType methodName(Parameters){  
Statements  
------  
------  
------  
}  
Call Method  
  
dataType variableName = methodName(values);  
  
Or  
System.out.println(methodname(values);  
------------------------------------------  
Example:  
   
package xyza;  
  
public class JavaMethods {  
  
//Create Method  
public static int multiply(int a, int b, int c){  
int result = a \* b \* c;  
return result;  
}  
  
public static void main (String [] args){  
//Call Method  
int x = multiply(10, 25, 35);  
System.out.println(x);  
      
System.out.println(multiply(10, 25, 35));  
}  
}  
-----------------------------------------  
2) Write method without returning any value  
  
a) Syntax for creating a Method (call the method by invoking Object)  
  
accessModifier returnTypeNothing methodName(parameters){  
Statements  
----------  
----------  
---------  
}  
  
//Create Object  
ClassName objectName = new ClassName();  
  
//Call Method  
object.method(values);  
-------------------------------------  
Example:  
   
public class JavaMethods {  
  
//Create Method  
public void studentRank(int marks){  
if (marks >= 600){  
System.out.println("Rank A");  
}  
else if (marks >= 500){  
System.out.println("Rank B");  
}  
else  
System.out.println("Rank C");  
}  
  
public static void main (String [] args){  
  
//Call method by invoking object  
JavaMethods obj = new JavaMethods();  
obj.studentRank(600);  
}  
}  
------------------------------  
2) Write method without returning any value  
b) Syntax for creating a Method (call the method without Object)  
   
accessModifier nonAccessModifier returnTypeNothing methoName(Parameters){  
Statements  
---------  
---------  
---------  
}  
Call method  
  
methodName(values);  
----------------------------------  
public class JavaMethods {  
  
//Create Method without returning any value (without object)  
public static void studentRank(int marks){  
if (marks >= 600){  
System.out.println("Rank A");  
}  
else if (marks >= 500){  
System.out.println("Rank B");  
}  
else{  
System.out.println("Rank C");  
}  
}  
  
public static void main (String [] args){  
  
//Call method without object  
studentRank(450);  
}  
}  
------------------------------------------  
**Usage of methods**  
   
a) Internal methods (defining/creating and calling methods within the same class)  
b) External methods (Calling methods from other classes)  
------------------------------------  
3a) Call External method (from another class) -by invoking Object  
Class 1:  
   
public class Sample1 {  
//Create Method  
public static int multiply(int a, int b, int c){  
int result = a \* b \* c;  
return result;  
}  
}//Create Method  
public static int multiply(int a, int b, int c){  
int result = a \* b \* c;  
return result;  
}  
}  
  
Class 2:   
  
  
public class Sample2 {  
//Create method  
public int add(int a, int b, int c){  
int result = a + b + c;  
return result;  
}  
      
public static void main (String [] args){  
  
//Create Object  
Sample2 obj1 = new Sample2();  
  
//Calling Internal method  
int x = obj1.add(10, 25, 35);  
System.out.println(x);  
  
//Create Object      
Sample1    obj2 = new Sample1();  
  
//Calling External Method  
int y = obj2.multiply(10, 25, 35);  
System.out.println(y);  
}  
}  
-----------------------------  
3b) Call External method (from another class) -without Object  
  
Class 1:  
   
public class Sample1 {  
  
//Create Method  
public static void studentRank(int marks){  
if (marks >= 600){  
System.out.println("Rank A");  
}  
else if (marks >= 500){  
System.out.println("Rank B");  
}  
else{  
System.out.println("Rank C");  
}  
}  
  
public static void main (String [] args){  
//Call method without object  
studentRank(450);  
}  
}  
  
Class 2:  
   
public class Sample2 {  
  
public static void main (String [] args){  
Sample1.studentRank(450);  
}  
}