FAST

National University of Computer and Emerging Sciences Peshawar

Lecture # 08

Software Construction and Development (Java Programming)

Instructor: Muhammad Abdullah Orakzai

DEPARTMENT OF COMPUTER SCIENCE

الذی علم بالقلم۔ علم الانسان ما لم یعلم۔

Functions / Methods in Java

Contents

1) Methods
2) Method Definition
3) Calling Method
4) Method Overloading
5) Tasks

Java Function/Methods

- Function is a set of instructions that are designed to perform a specific task.

- A function is a complete and independent program.

- It is executed by the main method to perform its tasks.

Java Function/Methods

- Functions are used to write the code of a large program by dividing it into smaller independent units.

- It avoids the replication of code in the program.

Functions VS Methods

Function — a set of instructions that perform a task.

Method — a set of instructions that are associated with an object.

METHODS

A method, like a function, is a set of instructions that perform a task. The difference is that a method is associated with an object, while a function is not.

Functions Types

1) **Built in Functions or standard library methods**

The standard library methods are built-in methods in Java that are readily available for use.

Example:

I. println()

II. nextInt()

III. showMessageDialog

IV. showInputDialog etc.

Functions Types

2) User Defined Functions

We can also create methods of our own choice to perform some task. Such methods are called user-defined methods.

Example:

```
public static void myMethod() {
System.out.println("My Function called");
}
```

Function definition

```
Access specifier    return type    methodName(list of parameter)
{
 statement(s);
}
```

The function definition is called method header.

Calling Method or Invoking Method

- Executing the statement(s) of method to perform task is called calling a function.

- Calling a method is called invoking a method.

Example:

```
addition();
```

Function have no parameters list and return type

```
Class Test
{
public void printStar()
 {
 System.out.println(“*****”);
 }
```

Function have no parameters list and return type…

```
public static void main(String args[])
{
Test object=new Test();
object.printStar();                    //method call
}
}   // Test Class body closed
```

Function have no return type but parameter list

```
Class Test1
{
public void sum(int x, int y)  // formal arguments
 {
 int sum=x+y;
 System.out.println(“Result is”+sum);
 }
```

Function have no return type but parameter list …

```
public static void main(String args[])
{
Test1 object=new Test1();
object.sum(5,6);     //Actual Arguments
}
}  // Test1 class body closed
```

Function return values

Function can return only one value.

Return Statement: The return statement is used to return calculated value from function definition to calling function.

Syntax:

```
return x;
```

Function return values…

```
Class Test2
{
public int sum(int x, int y)              // formal arguments
 {
 return(x+y);
 }
```

Function return values…

```
public static void main(String args[])
{
Test2 object=new Test2();
int result = object.sum(5,6);              //Actual Arguments
System.out.println(“Result is”+result);
}
}  // Test2 Class body closed
```

Function/Method Overloading

- Method having same name with different set of parameters (type, order, number) then such kind of method is called overloaded method and this mechanism is called method overloading.
- Method overloading is compile time polymorphism or static binding.
- It increase the readability of the program.

Note: In java method overloading is not possible by changing the return type of method.

Function/Method Overloading…

```
Class methodOverloading
{
public void sum(int x, int y)  // formal arguments
 {
 System.out.println(“sum of int is”+(x+y));
 }
public void sum(double x, double y)  // formal arguments
 {
 System.out.println(“sum of double is”+(x+y));
 }
```

Function/Method Overloading…

```
public void sum(int x, double y)  // formal arguments
 {
 System.out.println(“sum of int & double is”+(x+y));
 }

public void sum(double y, int x)  // formal arguments
 {
 System.out.println(“sum of double & int is”+(x+y));
 }
```

Function/Method Overloading…

```
public static void main(String args[])
{
methodOverloading object=new methodOverloading();
object.sum(3,5);
object.sum(3.3,5.6);
object.sum(3,5.4);
object.sum(3.6,5);
}
}
```

Why Method Overloading is not possible by changing the return type of method only?

In java, method overloading is not possible by changing the return type of the method only because of ambiguity. Let's see how ambiguity may occur:

```
class Adder{
static int add(int a,int b){return a+b;}
static double add(int a,int b){return a+b;}
}
```

Why Method Overloading is not possible by changing the return type of method only?...

```
class TestOverloading3{
public static void main(String[] args){
System.out.println(Adder.add(11,11));    //ambiguity


}


}
```

Function/Method Overloading…

Note: Method overloading is compile time polymorphism or static binding

Advantages of method overloading

It increase the readability of the program.

Note: In java method overloading is not possible by changing the return type of method.

Functions Tasks

1) Write function in java that will calculate table of a number in java. Number must be passed as argument to function parameters.

2) Write function in java that will find factorial of a number. Number must be passed as argument to function parameters.

3) Update your calculator using functions.

THANK YOU