**Notes: First you prepare how to write a code for syntax and write a code**

**Java: (day one)**

install java

install eclipse

create workspace

create project

File -> Project

We call Project is program

create .java file/class

Project -> new class and give extension .java

Class Employee {

}

**What is a Class ?**

In the real world, you'll often find many individual objects all of the same kind. There may be thousands of other bicycles in existence, all of the same make and model. Each bicycle was built from the same set of blueprints and therefore contains the same components. In object-oriented terms, we say that your bicycle is an *instance* of the *class of objects* known as bicycles. A *class* is the blueprint from which individual objects are created.

**What is an Object ?**

An object is a software bundle of related state and behavior. Software objects are often used to model the real-world objects that you find in everyday life.

**What Is a Package?**

A package is a namespace that organizes a set of related classes and interfaces. Conceptually you can think of packages as being similar to different folders on your computer. You might keep HTML pages in one folder, images in another, and scripts or applications in yet another. Because software written in the Java programming language can be composed of hundreds or thousands of individual classes, it makes sense to keep things organized by placing related classes and interfaces into packages.

The Java platform provides an enormous class library (a set of packages) suitable for use in your own applications. This library is known as the "Application Programming Interface", or "API" for short. Its packages represent the tasks most commonly associated with general-purpose programming.

**Definition:** A *package* is a grouping of related types providing access protection and name space management.

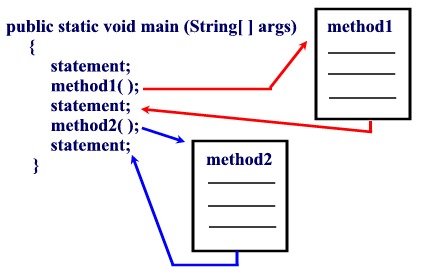
<https://docs.oracle.com/javase/tutorial/java/concepts/class.html>

**what is a method ?**

[**http://mathbits.com/MathBits/Java/Methods/Lesson1.htm**](http://mathbits.com/MathBits/Java/Methods/Lesson1.htm)

A method is a set of code which is referred to by name and can be called (invoked) at any point in a program simply by utilizing the method's name.  Think of a method as a subprogram that acts on data and often returns a value.

Each method has its own name.  When that name is encountered in a program, the execution of the program branches to the body of that method.  When the method is finished, execution returns to the area of the program code from which it was called, and the program continues on to the next line of code.



1. **how to create packages and what is best way to give name**

From solution explorer, select project, right click and select package

Ex: companyname.projectname.foldername (this is common naming standard)

<http://www.tutorialspoint.com/eclipse/eclipse_create_java_package.htm>

<https://docs.oracle.com/javase/tutorial/java/package/namingpkgs.html>

## Naming Conventions

Package names are written in all lower case to avoid conflict with the names of classes or interfaces.

Companies use their reversed Internet domain name to begin their package names—for example, com.example.mypackage for a package named mypackage created by a programmer at example.com.

Name collisions that occur within a single company need to be handled by convention within that company, perhaps by including the region or the project name after the company name (for example, com.example.region.mypackage).

Packages in the Java language itself begin with java. or javax.

In some cases, the internet domain name may not be a valid package name. This can occur if the domain name contains a hyphen or other special character, if the package name begins with a digit or other character that is illegal to use as the beginning of a Java name, or if the package name contains a reserved Java keyword, such as "int". In this event, the suggested convention is to add an underscore. For example:

|  |  |
| --- | --- |
| **Legalizing Package Names** | |
| **Domain Name** | **Package Name Prefix** |
| hyphenated-name.example.org | org.example.hyphenated\_name |
| example.int | int\_.example |
| 123name.example.com | com.example.\_123name |

1. **what is main method will do?**

Main method is starting point of program

A Java application must contain a main() method whose signature looks like this

public static void main(String args[])

The method signature for the main() method contains three modifiers:

* public indicates that the main() method can be called by any object. public, private, protected, and the implicit, friendly.
* static indicates that the main() method is a class method.  void indicates that the main() method has no return value.

The main() method in the Java language is similar to the main() function in C and C++. When you execute a C or C++ program, the runtime system starts your program by calling its main()function first. The main() function then calls all the other functions required to run your program. Similarly, in the Java language, when you execute a class with the Java interpreter, the runtime system starts by calling the class's main() method. The main() method then calls all the other methods required to run your application.

<http://www.cs.princeton.edu/courses/archive/spr96/cs333/java/tutorial/java/anatomy/main.html>

1. w**hat is data type and different data types**

It will represents what type of

Byte

Short

Int

Double

Float

Char

Boolean

<http://www.tutorialspoint.com/java/java_basic_datatypes.htm>

1. **creating property/data members**

int salary

1. **creating method with void**
2. **creating method with void and parameter**
3. **creating method with return data type**
4. **creating method with return data type and parameter**

[**http://www.tutorialspoint.com/java/java\_methods.htm**](http://www.tutorialspoint.com/java/java_methods.htm)

1. **creating variable:**

**Local Variable;**

**Instance Variables;**

**Class/Static Variables;**

**constructor :**

[**http://www.dummies.com/how-to/content/how-to-use-a-constructor-in-java.html**](http://www.dummies.com/how-to/content/how-to-use-a-constructor-in-java.html)

1. **creating static property**
2. **creating static method**
3. **creating object**
4. **calling method with no return**
5. **calling method with no return and parameter**
6. **calling method with return and no parameter**
7. **calling method with return and parameter**
8. **calling method with return and storing the return data**
9. **calling static method**
10. **using static property**

**Java: Day2**

1. **create classes under multiple packages**
2. **calling classes under different packages**
3. **write code to handle exceptions with try/catch/finally**
4. **what is final keyword**
5. **write code for interface and create class to implement that interface**
6. **write code for creating abstract class**
7. **implement method overloading**
8. **implement method overriding**
9. **implementing polymorphism**
10. **implementing interface**
11. **write a code to save data into excel file and read from excel file (POI and jexcel API)**
12. **how to update the data into XML file and read data from XML file**

**Java: Day3**

1. **write code to add items to integer, string array**
2. **write code to retrieve items from integer, string array**
3. **write code to add items to ArrayList collection**
4. **write code to retrieve items from arraylist**
5. **write code to add items HashMap**
6. **write code to retrieve items HashMap**
7. **write code to connect to JDBC to get rows from employee table**
8. **create Employee class**
9. **Add employee class to list collection**
10. **create method that return list of employee collection**