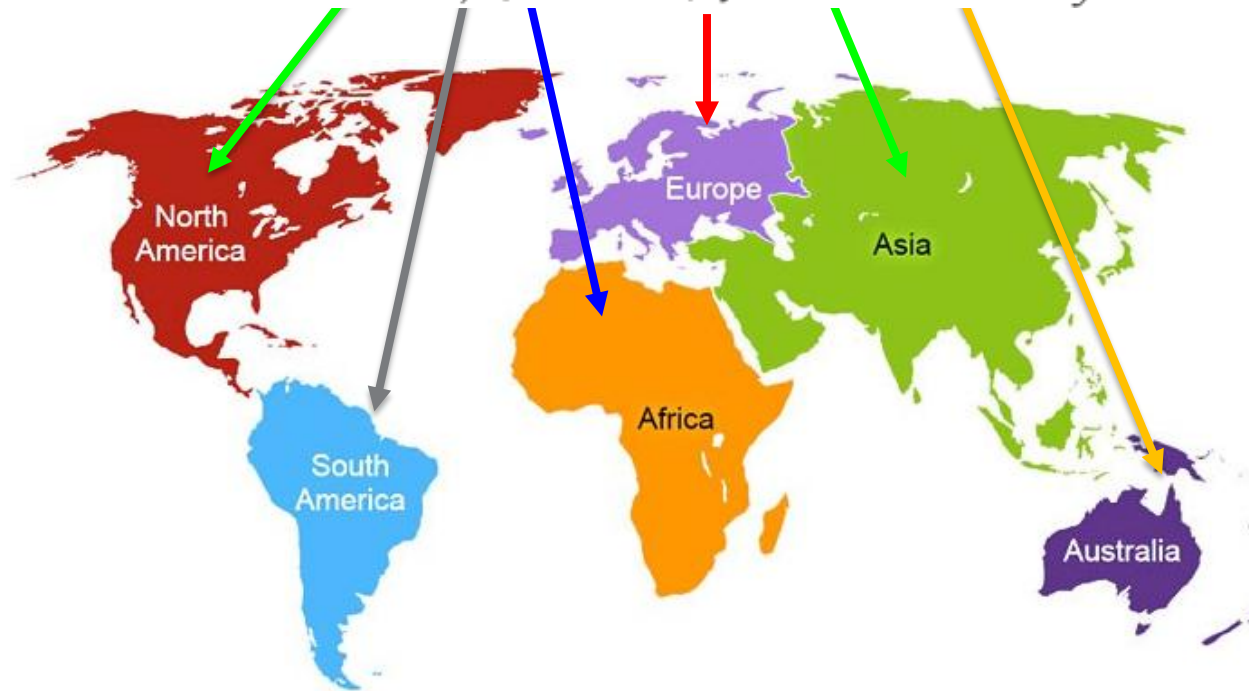





H2K Infosys, LLC
Dream, Strive & Achieve Victory



H2K Infosys is  business based in Atlanta, Georgia – United States
Providing Online IT training services world wide.

www.H2KINFOSYS.com

USA - +1-(770)-777-1269, UK - (020) 3371 7615
Training@H2KInfosys.com / H2KInfosys@Gmail.com

H2K INFOSYS PROVIDES WORLD CLASS SERVICES IN

**IT Trainings
with Real time
Project Work
for
Corporates
&
Individuals**

**Special IT
Training
for MS
Students in
US**

**Software Design,
Development,
QA – Manual,
Automation &
Performance
Testing,
Maintenance.**

**IT Staff
Augmentation**

**Job Placement
Assistance**

Tech Support

What are we learning today?

- **Understanding Access Modifiers**
- **Creating a Class**
- **Declaring Member variables**
- **Java Methods**
- **How to name methods**
- **Method Overloading**
- **Constructor**
- **Overloading Constructor**

Note: Things we are learning today will be used forever in Java Life.

- **How class is defined?**

```
class MyClass {  
    // field, constructor, and  
    // method declarations  
}
```

- **Simplest way to instantiate a Class:**

ClassName instanceName = new ClassName(); // Using 'new' keyword

- **Access Modifiers**

public modifier—the field / method is accessible from all classes.

private modifier—the field / method is accessible only within its own class.

protected modifiers – the field / method is accessible to package and subclasses.

No / default modifier – the field / method is accessible in package.

- **Declaring a method:**

More generally, method declarations have six components, in order:

- Modifiers—such as public, private, and others you will learn about later.
- The return type—the data type of the value returned by the method, or void if the method does not return a value.
- The method name—the rules for field names apply to method names as well, but the convention is a little different.
- The parameter list in parenthesis—a comma-delimited list of input parameters, preceded by their data types, enclosed by parentheses, (). If there are no parameters, you must use empty parentheses.
- An exception list—to be discussed later.
- The method body, enclosed between braces—the method's code, including the declaration of local variables, goes here.

- Java can distinguish between methods with different *method signatures*.
- Overloading is : methods within a class can have the same name if they have different parameter lists.
- For Example, Lets write a CustomerService.java class which has getCustomerList() method overloaded for following parameters:
 - Integer zipCode
 - String city
- Overloaded methods are differentiated by the number and the type of the arguments passed into the method
- You cannot declare more than one method with the same name and the same number and type of arguments, because the compiler cannot tell them apart.
- The compiler does not consider return type when differentiating methods, so you cannot declare two methods with the same signature even if they have a different return type.

A class contains constructors that are invoked to create objects from the class blueprint.

Constructor declarations look like method declarations—except that they use the name of the class and have no return type.

Lets create a Customer.java class which has a default Constructor for initialization.

You can overload Constructors same as methods. Let's write a overloading example.

Constructor is always there in class. **If you don't provide one, compiler will add it for you** (no-arg constructor)

Constructor also has access modifier constraints. **If another class cannot call a another class constructor, it cannot directly create that class's objects.**

A class that is derived from another class is called a subclass (also a derived class, extended class, or child class). The class from which the subclass is derived is called a superclass (also a base class or a parent class)..

Excepting Object, which has no superclass, every class has one and only one direct superclass (single inheritance). In the absence of any other explicit superclass, **every class is implicitly a subclass of Object**.

A subclass inherits all of the **public and protected** members of its parent, no matter what package the subclass is in.

A subclass **does not inherit the private** members of its parent class. However, if the superclass has public or protected methods for accessing its private fields, these can also be used by the subclass.

You can write a subclass constructor that invokes the constructor of the superclass, either **implicitly** or by using the keyword **super**.

Interesting: You can instantiate Subclass with Super Class declaration.

Let's create PreferredCustomer class which is Subclass of Customer class.

So now you can do this: `Customer pCustomer = new PreferredCustomer();`

With that rule anything in Java can be: `Object obj = new AnyClassNameHere();`

Thank you

Your feedback is highly important to improve our course material and teaching methodologies. Please email your suggestions.

**USA +1-(770)-777-1269
Training@H2KInfosys.com**

**UK (020) 3371 7615
H2KInfosys@Gmail.com**

H2K Infosys is e-Verified business based in Atlanta, Georgia – United States

H2K Infosys acknowledges the proprietary rights of the trademarks and product names of other companies mentioned in this document.



H2K Infosys, LLC (hereinafter “H2K”) acknowledges the proprietary rights of the trademarks and product names of other companies mentioned in any of the training material including but not limited to the handouts, written material, videos, power point presentations, etc. All such training materials are provided to H2K students for learning purposes only. H2K students shall not use such materials for their private gain nor can they sell any such materials to a third party. Some of the examples provided in any such training materials may not be owned by H2K and as such H2K does not claim any proprietary rights for the same. H2K does not guarantee nor is it responsible for such products and projects. H2K acknowledges that any such information or product that has been lawfully received from any third party source is free from restriction and without any breach or violation of law whatsoever.