**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.

**what is a method ?**

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.

**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)

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 |

**What a 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.

**Write Code for JDBC connection?**

package jdbcconnection;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class NewJdbc

{

public static void main(String[] args) throws SQLException, ClassNotFoundException {

String host="localhost";

String port="3306";

Connection c=DriverManager.getConnection("jdbc:mysql://"+ host +":" + port +"/sagar2","root","Vidy@123");

Statement s=c.createStatement();

ResultSet rs=s.executeQuery("Select\*from Room5211 where name='sagar'");

while(rs.next())

{

System.out.println(rs.getString("course"));

System.out.println(rs.getString("age"));

System.out.println(rs.getString("interviews"));

}

}

}

**Write code for adding elements to hashmap and typecasting using long?**

**package** javaprog;

**import** java.util.HashMap;

**public** **class** HashMapDemo {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

HashMap<String,Long> h=**new** HashMap<String,Long>();

//adding elements to hasmap and typecasting using long

h.put("fremont",(**long**) 94538);

h.put("dallas", (**long**) 94536);

h.put("hyderabad",(**long**) 500056);

h.put("secunderabad",(**long**) 500035);

}

}

**Write Code to retrieve elements from hashmap using key?**

**package** javaprog;

**import** java.util.HashMap;

**public** **class** HashMapDemo1 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

HashMap<String,Long> h=**new** HashMap<String,Long>();

//adding elements to hasmap and typecasting using long

h.put("fremont",(**long**) 94538);

h.put("dallas", (**long**) 94536);

h.put("hyderabad",(**long**) 500056);

h.put("secunderabad",(**long**) 500035);

//retrieve elements fromm hashmap using key

System.***out***.println(h.get("fremont"));

System.***out***.println(h.get("dallas"));

System.***out***.println(h.get("hyderabad"));

System.***out***.println(h.get("secunderabad"));

}

}

**Write code to add items to HashSet?**

**package** javaprog;

**import** java.util.HashSet;

**public** **class** HashSetDemo {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

//creating a HashSet to store Strings

HashSet<String> hs=**new** HashSet<String>();

//add items to HashSet

hs.add("apple");

hs.add("ball");

hs.add("cat");

hs.add("dog");

hs.add("elephant");

}

}

**Write code to retrieve elements from hashset?**

**package** javaprog;

**import** java.util.HashSet;

**public** **class** HastSetDemo1 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

//creating a HashSet to store Strings

HashSet<String> hs=**new** HashSet<String>();

//add items to HashSet

hs.add("apple");

hs.add("ball");

hs.add("cat");

hs.add("dog");

hs.add("elephant");

//retrieve elements from hashset

System.***out***.println(hs);

}

}

**Write Code for creating Method with return data type?**

**package** javaprog;

**public** **class** methodreturn {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

methodreturn m= **new** methodreturn();

m.Ball("shape is circle");

m.Box();

System.***out***.println(m.prime());

}

**public** **void** Box()

{

System.***out***.println("shape of box is rectangle");

}

**public** **int** prime(){

**return** 3;

}

**public** **void** Ball(String s)

{

System.***out***.println(s);

}

}

**Write code for Method overriding?**

**package** javaprog;

**public** **class** overiding {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Student s=**new** Student();

Student k=**new** qa();

s.name();

k.name();

}

}

**class** Student{

//this is first method

**public** **void** name(){

System.***out***.println("Vidya sagar");

}

}

**class** qa **extends** Student{

**public** **void** name(){

System.***out***.println("sama");

}

}

**Write code for method overloading?**

**package** javaprog;

**public** **class** overiding {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Student s=**new** Student();

Student k=**new** qa();

s.name();

k.name();

}

}

**class** Student{

//this is first method

**public** **void** name(){

System.***out***.println("Vidya sagar");

}

}

**class** qa **extends** Student{

**public** **void** name(){

System.***out***.println("sama");

}

}

**Write Code for Constructor?**

**package** javaprog;

**public** **class** constructor {

**public** constructor(){

System.***out***.println("constructor is created");

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

constructor c= **new** constructor();

}

}

**Write Code for Constructor Overloading?**

**package** javaprog;

**public** **class** constuctorovrloading {

**public** constuctorovrloading(**int** a){

System.***out***.println(a);

}

**public** constuctorovrloading(**int** a,**int** b){

System.***out***.println(6\*a\*b);

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

constuctorovrloading c=**new** constuctorovrloading(21);

constuctorovrloading c1=**new** constuctorovrloading(10,20);

}

}

**Write Code using getset method?**

**package** javaprog;

**public** **class** getset {

**private** **int** rollnum;

**private** String studentname;

**public** **void** setID(**int** a){

rollnum=a;

}

**public** **int** getID(){

**return** rollnum;

}

**public** **void** setName(String s){

studentname=s;

}

**public** String getName(){

**return** studentname;

}

**public** **static** **void** main(String[]args) {

// **TODO** Auto-generated method stub

getset g=**new** getset();

System.***out***.println("student name is sagar:"+g.getName());

System.***out***.println("rollnum is 200:"+ g.getID());

}

}

**Write Code to add elements using ArrayList?**

**package** javaprog;

**import** java.util.ArrayList;

**public** **class** ArrayListDemo {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

//create ArrayList

ArrayList<String>ar=**new** ArrayList<String>();

//add objects

ar.add("fremont");

ar.add("dallas");

ar.add("atlanta");

ar.add("sanjose");

}

}

**Write code for retrieving data from ArrayList?**

**package** javaprog;

**import** java.util.ArrayList;

**public** **class** ArrayListDemo1 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

ArrayList<String>ar=**new** ArrayList<String>();

//add objects

ar.add("fremont");

ar.add("dallas");

ar.add("atlanta");

ar.add("sanjose");

//display elements using for each

**for**(String i:ar)

{

System.***out***.println(i);

}

}

}

**Write Code to add two strings?**

**package** javaprog;

**public** **class** String1 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

String s1="data";

String s2="base";

s1=s1+s2;

System.***out***.println(s1);

}

}

**Write Code using Sting Buffer?**

**package** javaprog;

**public** **class** StringBuffer1 {

**public** **static** **void** main(String[] args)

{

StringBuffer sb=**new** StringBuffer();

String sur="sama";

String middle="vidya";

String last="sagar";

sb.append(sur);

sb.append(last);

System.***out***.println("name = " +sb);

**int** n= sur.length();

System.***out***.println(n);

sb.insert(n, middle);

System.***out***.println("Full Name = " +sb);

}

}

**Write Code using Sting Buffer?**

**public** **void** run() {

**for**(**int** i=1;i<=10;i++)

{

System.***out***.println(str+" "+i);

**try**{

Thread.*sleep*(2000);

}

**catch**(InterruptedException ie)

{

ie.printStackTrace();

}

}

}}

**public** **class** StringBufferDemo {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

StringBuffer one=**new** StringBuffer("vidya");

StringBuffer two=**new** StringBuffer("sagar");

MyThread1 sb=**new** MyThread1(one);

MyThread1 sb1=**new** MyThread1(two);

Thread t1= **new** Thread(sb);

Thread t2= **new** Thread(sb1);

t1.start();

t2.start();

}

}

**Write Code using String Bulider?**

**package** javaprog;

**class** MyThread **implements** Runnable {

StringBuilder str;

MyThread(StringBuilder str)

{

**this**.str=str;

}

**public** **void** run() {

**for**(**int** i=1;i<=10;i++)

{

System.***out***.println(str+" "+i);

}

}}

**class** StringBuilder1

{

**public** **static** **void** main(String args[])

{

StringBuilder one=**new** StringBuilder("vidya");

StringBuilder two=**new** StringBuilder("sagar");

MyThread sb=**new** MyThread(one);

MyThread sb1=**new** MyThread(two);

Thread t1= **new** Thread(sb);

Thread t2= **new** Thread(sb1);

t1.start();

t2.start();

}

}