|  |
| --- |
| //  //Program to understand constructor  package crt;  class Test  {  int add(int a,int b)  {  int c=a+b;  return c;  }  Test()  {  System.out.println("Default Constructor");  }  Test(int x,int y)  {  System.out.println("Parameterized constructor:"+"X="+x+" Y="+y);  }  }  public class Constructor {  public static void main(String[] args) {  // TODO Auto-generated method stub  Test t=new Test();  int res=t.add(5,10);  System.out.println("Add method called:"+res);  Test t1=new Test(5,10);  res=t1.add(5,10);  System.out.println("Add Method called"+res);  }  } |
| //Program to understand memory allocation for variable  **package** crt;  **public** **class** Variable {  **public** **static** **void** main(String args[])  {  **int** a;//in java memory for variable is created only when value is assigned to variable  **int** b=20;  //System.out.println("a="+a+" b="+b);  }  } |
| //Program to understand Arguments and parameters  **package** crt;  **class** Test1  {  **void** add(**int** a,**int** b)//Parameters:copied values of Arguments  {  **int** c=a+b;  System.***out***.println(c);  }  }  **public** **class** Args\_Param {  **public** **static** **void** main(String args[])  {  Test1 t=**new** Test1();  t.add(87,102);//Arguments:Real values  }  } |
| **package** crt;  **class** Class\_variable //Class variable can be declared both in class and constructor  {  **static** **int** *a*=10;  Class\_variable()  {  **int** b=20;  System.***out***.println(b );  }  }  **class** Static\_variable //Static variable can be declared inside class only  {  **static** **int** *a*=100;    }  **class** Instance\_variable  {  **int** a=1000;  }  **public** **class** Variable\_Scope {  **public** **static** **void** main(String[] args) {  // **TODO** Auto-generated method stub  Class\_variable ob=**new** Class\_variable();  System.***out***.println(Class\_variable.*a*);  System.***out***.println(Static\_variable.*a*);  Instance\_variable obj=**new** Instance\_variable();  System.***out***.println(obj.a);  }  } |
| **package** crt;  **public** **class** Constructor\_overload {  Constructor\_overload(**int** a,**int** b)  {  **int** c=a+b;  System.***out***.println("Sum:"+c);  }  Constructor\_overload(**double** a,**double** b)  {  **double** c=a+b;  System.***out***.println("Sum:"+c);  }  Constructor\_overload(String a,String b)  {  String c=a+b;  System.***out***.println("concatenated string:"+c);  }  **public** **static** **void** main(String args[])  {  Constructor\_overload obj=**new** Constructor\_overload(10,20);  Constructor\_overload obj1=**new** Constructor\_overload(10.5,20.5);  Constructor\_overload obj2=**new** Constructor\_overload("Hii ","Nagu");  }  } |
| **package** crt;  **public** **class** Method\_overload {  **void** Method\_overload(**int** a,**int** b)  {  **int** c=a+b;  System.***out***.println("Sum:"+c);  }  **void** Method\_overload(**double** a,**double** b)  {  **double** c=a+b;  System.***out***.println("Sum:"+c);  }  **void** Method\_overload(String a,String b)  {  String c=a+b;  System.***out***.println("concatenated string:"+c);  }  **public** **static** **void** main(String args[])  {  Method\_overload obj=**new** Method\_overload();  obj.Method\_overload(10, 20);  obj.Method\_overload(10.5, 20.5);  obj.Method\_overload("Hii ", "Nagu");  }  } |
| **package** crt;  **public** **class** Method\_override {  **public** **void** eat()  {  System.***out***.println("Eating");  }  **public** **static** **void** main(String[] args) {  // **TODO** Auto-generated method stub    Dog g=**new** Dog();  g.eat();  Cat c=**new** Cat();  c.eat();  Method\_override m=**new** Method\_override();  m.eat();  }  }  **class** Dog **extends** Method\_override  {  **public** **void** eat()  {  System.***out***.println("Dog is eating");  }  }  **class** Cat **extends** Method\_override  {  **public** **void** eat()  {  System.***out***.println("Cat is eating");  }  } |
| **package** crt;  **public** **class** Multilevel\_Inheritance {  **public** **void** hii()  {  System.***out***.println("Inside the parent class");  }  **public** **static** **void** main(String args[])  {  C1 c1=**new** C1();  c1.hii();  C2 c2=**new** C2();  c2.hii();  c2.C1();    }  }  **class** C1 **extends** Multilevel\_Inheritance  {  **void** C1()  {  System.***out***.println("Inside class C1");  }  }  **class** C2 **extends** C1  {  C2()  {  System.***out***.println("Inside class C2");  }  } |
| **package** crt;  **public** **class** Hierarchical {  **public** **void** greet()  {  System.***out***.println("Inside Parent class");  }  **public** **static** **void** main(String[] args) {  // **TODO** Auto-generated method stub  Ch1 c1=**new** Ch1();  c1.greet();  Ch2 c2=**new** Ch2();  c2.greet();    }  }  **class** Ch1 **extends** Hierarchical  {  Ch1()  {  System.***out***.println("Inside class C1");  }  }  **class** Ch2 **extends** Hierarchical  {  Ch2()  {  System.***out***.println("Inside class C2");  }  } |
| **package** crt;  **public** **class** Hybrid {  **void** eat()  {  System.***out***.println("Eating");  }  **public** **static** **void** main(String[] args) {  // **TODO** Auto-generated method stub  A a=**new** A();  a.run();  a.eat();  B b=**new** B();  b.run();  b.eat();  }  }  **interface** Runnable  {  **abstract** **void** run();//abstract void run() and void run() both are the same  }  **class** A **extends** Hybrid **implements** Runnable  {  **public** **void** run()  {  System.***out***.println("Inside A run method implemented");  }  }  **class** B **extends** Hybrid **implements** Runnable  {  **public** **void** run()  {  System.***out***.println("Inside B run method implemented");  }  } |
| **//Within the package both public and protected methods works in the same way but when try to access in other package it is must and should to extend the class that contains protected method**  **package crt;**  **public class Ex\_public {**  **public void play()**  **{**  **System.*out*.println("Playing");**  **}**  **public static void main(String[] args) {**  **// TODO Auto-generated method stub**  **Hii h=new Hii();**  **h.hii();**  **h.test();**  **}**  **}**  **class Hii**  **{**  **public void hii()**  **{**  **System.*out*.println("Inside Hii");**  **}**    **void test()**  **{**  **Ex\_public a=new Ex\_public();**  **a.play();**  **}**  **}** |
| **package basic;**  **import crt.Ex\_public;**  **public class Ex\_public\_child {**  **public void test()**  **{**  **Ex\_public e=new Ex\_public();**  **e.play();**  **}**  **public static void main(String[] args) {**  **// TODO Auto-generated method stub**  **Ex\_public\_child cc=new Ex\_public\_child();**  **cc.test();**  **}**  **}** |
| **//Within the package both public and protected methods works in the same way but when try to access in other package it is must and should to extend the class that contains protected method**  **package basic;**  **public class Ex\_Protected {**  **protected void play()**  **{**  **System.*out*.println("Playing");**  **}**  **public static void main(String[] args) {**  **// TODO Auto-generated method stub**  **Hiii h=new Hiii();**  **h.hii();**  **h.test();**  **}**  **}**  **class Hiii**  **{**  **public void hii()**  **{**  **System.*out*.println("Inside Hii");**  **}**    **void test()**  **{**  **Ex\_Protected a=new Ex\_Protected();**  **a.play();**  **}**  **}** |
| **//Within the package both public and protected methods works in the same way but when try to access in other package it is must and should to extend the class that contains protected method**  **package crt;**  **import basic.Ex\_Protected;**  **public class Ex\_protected\_child extends Ex\_Protected{**  **public void test()**  **{**  **play();**  **}**  **public static void main(String args[])**  **{**  **Ex\_protected\_child c=new Ex\_protected\_child();**  **c.test();**  **}**  **}** |
| **package crt;**  **abstract public class Ex\_Abstract {**  **abstract void add(int a,int b);**  **int sub(int a,int b)**  **{**  **int c=a-b;**  **return c;**  **}**  **public static void main(String args[])**  **{**  **Testt t=new Testt();**  **t.add(10, 10);**  **int res=t.sub(10, 5);//Abstract classes cant be instantiated so use child reference to access parent methods**  **System.*out*.println("Subtraction:"+res);**  **}**  **}**  **class Testt extends Ex\_Abstract**  **{**  **void add(int a,int b)**  **{**  **int c=a+b;**  **System.*out*.println("Addition:"+c);**  **}**  **}** |
| **package crt;**  **public class Ex\_String {**  **public static void main(String[] args) {**  **// TODO Auto-generated method stub**  **String s=new String("Hello");**  **String a="Hii";**  **String b="Hii";//String Constant Pool maintains single copy,Both a and b points to that address**  **String s1=new String("Hello");//By using new keyword new memory block will be created**  **if(s==s1)//Memory location**  **{**  **System.*out*.println("S and S1 are in same location");**  **}**  **else**  **{**  **System.*out*.println("S and S2 are not in same location");**  **}**  **if(s.equals(s1))//Data comparision**  **{**  **System.*out*.println("S and S1 contains same data");**  **}**  **else**  **{**  **System.*out*.println("S and S2 do not contains same data");**  **}**  **if(a==b)//Memory location**  **{**  **System.*out*.println("A and B are in same location");**  **}**  **else**  **{**  **System.*out*.println("A and B are not in same location");**  **}**  **if(a.equals(b))//Data comparision**  **{**  **System.*out*.println("A and B contains same data");**  **}**  **else**  **{**  **System.*out*.println("A and B do not contains same data");**  **}**  **}**  **}** |