**ASSIGNMENT 2 : CONSTRUCTORS**

**1 .Program about java constructor**

* **public** **class** Main {

**private** String name;

//Constructor

Main()

{

System.***out***.println("constructor called:");

name="programiz";

}

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

//constructor is invoked while creating an object of the Main class

Main obj=**new** Main();

System.***out***.println("the name is " +obj.name);

}

}

**Output:**

constructor called:

the name is programiz

java private no arg constructor

* **class** Main1 {

**int** i;

// constructor with no parameter

**private** Main1()

{

i=5;

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

}

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

//calling the constructor without any parameter

Main1 obj=**new** Main1();

System.***out***.println("Value of i: "+obj.i);

}

}

**Output:**

constructor is called

Value of i: 5

**2 .java public no argument constructor**

**class** company {

String name;

//public constructor

**public** company(){

name="programiz";

}

}

**class** Main2

{

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

//object is created in another class

company obj=**new** company();//calling the constructor without parameter

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

}

}

**Output:**

company name =programiz

**3 .parameterized constructor**

**class** Main3 {

String languages;

//single parametrized constructor

Main3(String lang)

{

languages=lang;

System.***out***.println(languages+ "programming language");

}

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

//calling constructor by passing single value

Main3 obj1=**new** Main3("java");

Main3 obj2=**new** Main3("python");

Main3 obj3=**new** Main3("c");

}

}

**Output:**

javaprogramming language

pythonprogramming language

cprogramming language

**4.program for default constructor**

**class** Main4 {

//instance variables

**int** a;

**boolean** b;

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

Main4 obj=**new** Main4();//default constructor is called

System.***out***.println("Default values");

System.***out***.println("a = "+obj.a);

System.***out***.println("b ="+obj.b);

}

}

**Output:**

Default values

a = 0

b =false

**5 .java constructor overloading**

**class** Main5 {

String langauge;

//constructor with no parameter

Main5()

{

**this**.langauge="java";

}

//constructor with single parameter

Main5(String langauge)

{

**this**.langauge=langauge;

}

**public** **void** getName()

{

System.***out***.println("programming language: "+**this**.langauge);

}

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

//call constructor with no parameter

Main5 obj1=**new** Main5();

//call constructor with single parameter

Main5 obj2=**new** Main5("python");

obj1.getName();

obj2.getName();

}

}

Output:

programming language: java

programming language: python

**6 .Java Program to create and call a default constructor**

**class** Bike1{

//creating a default constructor

Bike1(){System.***out***.println("Bike is created");}

//main method

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

//calling a default constructor

Bike1 b=**new** Bike1();

}

}

Output:

Bike is created

**7 .Let us see another example of default constructor**

**which displays the default values**

**class** Student3{

**int** id;

String name;

//method to display the value of id and name

**void** display()

{

System.***out***.println(id+" "+name);

}

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

{

//creating objects

Student3 s1=**new** Student3();

Student3 s2=**new** Student3();

//displaying values of the object

s1.display();

s2.display();

}

}

**Output:**

0 null

0 null

**8 .Java Program to demonstrate the use of the parameterized constructor.**

**class** Student4{

**int** id;

String name;

//creating a parameterized constructor

Student4(**int** i,String n){

id = i;

name = n;

}

//method to display the values

**void** display(){System.***out***.println(id+" "+name);}

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

//creating objects and passing values

Student4 s1 = **new** Student4(111,"Karan");

Student4 s2 = **new** Student4(222,"Aryan");

//calling method to display the values of object

s1.display();

s2.display();

}

}

**Output:**

111 Karan

222 Aryan

**9 .Java program to overload constructors**

**class** Student5{

**int** id;

String name;

**int** age;

//creating two arg constructor

Student5(**int** i,String n){

id = i;

name = n;

}

//creating three arg constructor

Student5(**int** i,String n,**int** a){

id = i;

name = n;

age=a;

}

**void** display(){System.***out***.println(id+" "+name+" "+age);}

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

Student5 s1 = **new** Student5(111,"Karan");

Student5 s2 = **new** Student5(222,"Aryan",25);

s1.display();

s2.display();

}

**}**

**Output**:

111 Karan 0

222 Aryan 25

**10 . Java Program to illustrate calling a no argument constructor**

**import** java.io.\*;

**class** Geek

{

**int** num;

String name;

// this would be invoked while an object

// of that class is created.

Geek()

{

System.***out***.println("Constructor called");

}

}

**class** GFG

{

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

{

// this would invoke default constructor.

Geek geek1 = **new** Geek();

// Default constructor provides the default

// values to the object like 0, null

System.***out***.println(geek1.name);

System.***out***.println(geek1.num);

}

}

**Output:**

Constructor called

null

0

**11 . Java Program to Illustrate Working of parameterized constructor**

// Class 1

**class** class1 {

// data members of the class.

String name;

**int** id;

// Constructor would initialize data members

// With the values of passed arguments while

// Object of that class created

class1(String name, **int** id)

{

**this**.name = name;

**this**.id = id;

}

}

**class** class2 {

// main driver method

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

{

// This would invoke the parameterized constructor.

class1 geek1 = **new** class1("adam", 1);

System.***out***.println("GeekName :" + geek1.name

+ " and GeekId :" + geek1.id);

}

}

**Output:**

GeekName :adam and GeekId :1

**12 .** **Java Program to illustrate constructor overloading using same task (addition operation ) for different,types of arguments.**

**class** Geek2

{

// constructor with one argument

Geek2(String name)

{

System.***out***.println("Constructor with one " +

"argument - String : " + name);

}

// constructor with two arguments

Geek2(String name, **int** age)

{

System.***out***.println("Constructor with two arguments : " +

" String and Integer : " + name + " "+ age);

}

// Constructor with one argument but with different

// type than previous..

Geek2(**long** id)

{

System.***out***.println("Constructor with one argument : " +

"Long : " + id);

}

}

**class** class3

{

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

{

// Creating the objects of the class named 'Geek'

// by passing different arguments

// Invoke the constructor with one argument of

// type 'String'.

Geek2 geek2 = **new** Geek2("Shikhar");

// Invoke the constructor with two arguments

Geek2 geek3 = **new** Geek2("Dharmesh", 26);

// Invoke the constructor with one argument of

// type 'Long'.

Geek2 geek4 = **new** Geek2(325614567);

}

}

**Output:**

Constructor with one argument - String : Shikhar

Constructor with two arguments : String and Integer : Dharmesh 26

Constructor with one argument : Long : 325614567