**Constructor–Assignment**

Question 1: -**What is a Constructor?.**

Ans: -A constructor is a special method of a class in object-oriented programming that initializes a new created object of that type. Whenever an object is created, the constructor is called automatically.It doesn’t have a return type and it cannot be static, final, abstract, and synchronized.

Question 2: -**What is a Constructor chaining ?.**

Ans: -Constructor chaining is the process of calling a sequence of constructors.

We can do it in two ways:

* by using *this()* keyword for chaining constructors in the same class
* by using *super()* keyword for chaining constructors from the parent class

Question 3: -**Can we call a subclass Constructor from a superclass constructor?.**

Ans: -No,we can not call a subclass constructor from a superclass constructor.

Ex: -

**package** hello;

**class** hello {

// Superclass constructor

hello() {

System.***out***.println("hello PWSkill");

// this is compile time error occurred,so this the reason we can't call

// subconstructor from super constructor

constructor();

}

}

**public** **class** constructor **extends** hello {

// subclass constructor

**public** constructor() {

System.***out***.println("bye PWSkill");

}

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

System.***out***.println("constructor calling");

}

}

Question 4: -**What happens if you keep a return type for a Constructor?.**

Ans: -No,we can not keep return type for a constructor,if we keep return type for constructor then this constructor reacts a normal method .

**Ex: -**

**package hello;**

**public class constructor {**

**String constructor() {**

**System.*out*.println("bye PWSkill");**

**return "return method";**

**}**

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

**constructor con=new constructor();**

**con.constructor();**

**System.*out*.println("constructor calling");**

**}**

**}**

Question 5: -**What is Non-arg Constructor?.**

Ans: -Constructor without any argument is called a no-args constructor. It’s like overriding the default constructor .

**Ex: -**

**package hello;**

**public class constructor {**

**//Non-args Constructor**

**public constructor() {**

**System.*out*.println("Non-args constructor");**

**}**

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

**constructor con=new constructor();**

**;**

**}**

**}**

Question 6: -**How is a No-argument Constructor different from the default constructor?.**

Ans: -Default constructors are sometimes called no-arg constructors since they both work the same. But no-arg constructor is created by the **user** while default constructor can only be created by the **compiler**.

Question 7: -**When do we need constructor overloading?.**

Ans: -The constructors are overloaded to initialize the objects of a class in different ways. This allows us to initialize the object with either default values or used given values. If data members are not initialized then program may give unexpected results

Question 8: -**When is the default constructor Explain with an example?.**

Ans: -A *default constructor* is a constructor that either has no parameters, or if it has parameters, *all* the parameters have default values.

**Ex: -**

**package** hello;

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

**int** fees;

String name;

**double** duration;

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

constructor con=**new** constructor();

System.***out***.println(con.fees);

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

System.***out***.println(con.duration);

}

}