What is a Constructor? Syntax of creating a constructor.

A constructor is a **special method** of the class that is called when an instance of the class is created i.e. when we create object with new keyword then the constructor is called. They are used to give initial values to variables of class.

Syntax:

```
<class_name>()               //constructor name should match the class name
{
    //body of constructor
    //give values to variables of class here
}
```

Example 1: A Simple Constructor that takes no input

```
class Point
{
    int x;
    int y;

    Point()
    {
          System.out.println("Inside Constructor");
    }
}
```

Calling a constructor

Constructors are called using new keyword.

Syntax:

```
new class_name();
```

Example 2: Calling our constructor from Example 1

```
Point p1 = new Point();                          //Point() here is constructor
```

Output:

```
Inside Constructor
```

Concept of Default Constructor

When you do not specify a constructor in the class, a default constructor is automatically created in the class. Default constructor provides the default values to the member variables like 0 for integer variables, null for Strings etc. depending on the type.

Example 3: Using default Constructor

```
1  class Student
2  {
3     String name;
4     int marks;
5  }
```

Code:

```
1  Student st = new Student();                    //default constructor is invoked
2  System.out.println(st.name);
3  System.out.println(st.marks);
```

Output:

```
null
0
```

Defining Default Constructor

You can give your own definition to default constructor. Just like we did in Example 1.

Example 4: Defining default Constructor

```
1  class Point
2  {
3     int x;
4     int y;
5
6     Point()
7     {
8           x = 4;
9           System.out.println("inside default constructor");
10    }
11 }
```

Code:

```
1  Point p1 = new Point();                        //default constructor is invoked
2  System.out.println("Point1 :: " + p1.x + " : " + p1.y);
3  Point p2 = new Point();
4  p2.y = 10;
5  System.out.println("Point2 :: " + p2.x + " : " + p2.y);
```

Output:

```
inside default constructor
Point1 :: 4 : 0
inside default constructor
Point2 :: 4 : 10
```

Note: The constructor of a class is invoked every time an object is created. In above example two point objects are created.

Problem 1	Give answers for each of the following code segments. Note: Explanation of each answer is mandatory.
Problem 1.1	**Refer the Class given below and give output of code that follows.**

Class

```
1  class Person
2  {
3      String name;
4      int age;
5  }
```

Code

```
1  Person p=new Person();
2  System.out.println("Name = "+ p.name + " age = " + p.age);
```

Problem 1.2	Refer the Class given below and give output of code that follows.

Class

```
1  class Person
2  {
3      String name;
4      int age;
5
6      Person()
7      {
8             System.out.println("Non Parameterized constructor");
9      }
10 }
```

Code

```
1  Person p = new Person();
```

Problem 1.3	Refer the Class given below and give output of code that follows.

Class

```
1  class Person
2  {
3      String name;
4      int age;
5
6      Person()
7      {
8             name = "";
9             age = 0;
10     }
11 }
```

Code

```
1  Person p = new Person();
2  System.out.println("Name = "+ p.name + " age = " + p.age);
```

Parameterized constructor

A constructor that takes input is known as parameterized constructor.

Syntax:

```
1  class_name(data-type variable1 , data-type variable2)
2  {
3     //constructor body
4  }
```

Example 5: Using Parameterized constructor

```
1  class Student
2  {
3     int rollNo;
4     String name;
5
6     Student(int x, String str)
7     {
8           rollNo = x;
9           name = str;
10    }
11 }
```

Code:

```
1  Student s1 = new Student(111, "Kunal");
2  Student s2 = new Student(222, "Aryan");
3  System.out.println(s1.rollNo + " : " + s1.name);
4  System.out.println(s2.rollNo + " : " + s2.name);
```

Output:

```
111 : Kunal
222 : Aryan
```

Parameterized constructor with Default

Example 6: Using default with parametrized constructor

```
1  class Student
2  {
3     String name;
4
5     Student(String str)
6     {
7           name = str;
```

```
8      }
9   }
```

Code:

```
1   Student s1 = new Student();         //error in this line (read explanation)
2   Student s2 = new Student("Rakesh");
3   Student s3 = new Student(200);      //error in this line (read explanation)
```

Output:

```
Compiler Error in line 1
Compiler Error in line 3
```

Explanation:

The error in line 1 is due to non-availability of default constructor. To rectify this error either remove this line or add a default constructor yourself.

Error in line 3 is because we haven't defined any constructor that takes int as input.

Rules of using default constructor:

- **If no constructor is defined:** Then a default constructor is provided by java itself. And you can use it.
- **If any parameterized constructor is defined:** Then you can use only parameterized constructor. Default is not available. If you want to use both then you have to create a default constructor yourself.

Rules of using parameterized constructor:

- If you have defined parameterized constructor then object can be created only if we pass right number of and correct type of values.
- If type or number of arguments (values) do not match with any constructor then compile time error will occur.

Problem 2	**Give answers for each of the following code segments.** **Note: Explanation of each answer is mandatory.**
Problem 2.1	**Refer the Class given below and give output of code that follows.**
Class	

```
1   class Laptop
2   {
3       String type;
4       Laptop(String str)
5       {
6               System.out.println("Parameterized Constructor");
7       }
8   }
```

Code

```
1  Laptop lp1 = new Laptop("Celeron");
2  Laptop lp2 = new Laptop();
```

Problem 2.2 **Refer the Class given below and give output of code that follows.**

Class

```
1  class Laptop
2  {
3     String type;
4     Laptop(String str)
5     {
6           System.out.println(str);
7           type = str;
8     }
9  }
```

Code

```
1  Laptop lp1 = new Laptop("Celeron");
2  System.out.println(lp1.type);
```

Problem 2.3 **Refer the Class given below and give output of code that follows.**

Class

```
1  class Person
2  {
3     String name;
4     int age;
5
6     Person(String name1, int age1)
7     {
8           name = name1;
9           age = age1;
10    }
11 }
```

Code

```
1  Person p1 = new Person("Messi", 28);
2  Person p2 = new Person("Max");
3  Person p3 = new Person();
4  p3.name = "Antonio";
5  p3.age = p2.age;
6  System.out.println("Name = " + p3.name + " age = " + p3.age);
```

Problem 3	For each of the following code segments write code that satisfies the desired output. Note: Explanation of each answer is mandatory.

Problem 3.1 Replace //INSERT HERE with your code to generate mentioned output.

Class

```
1  class Laptop
2  {
3     String type;
4
5     Laptop(String str)
6     {
7           //INSERT HERE
8     }
9  }
```

Code	Output
`1  Laptop lp1 = new Laptop("Dell");` `2  System.out.println(lp1.type);`	`Dell Laptop` `Dell`

Problem 3.2 Replace //INSERT HERE with your code to generate mentioned output.

Class

```
1  class Laptop
2  {
3     String type;
4     int price;
5
6     Laptop(String str, int price1)
7     {
8           //INSERT HERE
9     }
10
11    Laptop()
12    {
13          //INSERT HERE
14    }
15 }
```

Code	Output

`1  Laptop lp1 = new Laptop();`
`2  System.out.println(lp1.type);`
`3  System.out.println(lp1.price);`
`4  Laptop lp2 = new Laptop("HCL", 50000);`
`5  System.out.println(lp2.type);`
`6  System.out.println(lp2.price);`
	`Default Brand`
	`null`
	`0`
	`Brand Specified`
	`HCL`
	`50000`

Problem 4.1	**Create a class Item with two variable names quantity and price.**
Item	
Problem 4.2	**Create constructor Item(int quantity1, int price1). Use this constructor to give values to class variables quantity and price.**
Problem 4.3	**Create default constructor that initialize price with 10 and quantity with 5.**
Problem 4.4	**Use default constructor to create an object. Create another object with quantity 20 and price 10 by using constructor created in Problem 4.2. Print values of both objects.**