Keypoints in Java Constructor:

1) Constructor name should be same as class name.
2) Constructor should not be having any return type.
3) We need to initialize a class object with the available constructors present in our program.
4) In a Java program if we are not writing any constructors i.e, either parameterized constructor or non-parameterized constructor, then Java compiler is going to provide our program with a default constructor.
5) default constructor is same like non parameterized constructor.

default constructor ⟹ Provided by the Compiler
non parameterized constructor ⟹ Provided by the Programmer

6) The access modifier for the default constructor provided by the compiler will be SAME as the access modifier of class. (If the class is public then constructor is also public OR If the class is default then constructor is also default).
7) If we declare a constructor as 'private' then we can restrict the object creation of our class in other classes.
8) Just like void methods inside a constructor also we can write a return statement without returning anything.
9)

Constructor Overloading is Possible
Constructor Overridding is NOT Possible

Key points:

- Constructor can have all '4' access modifiers (public, protected, private, default).
- If there is no constructor in a class, compiler automatically creates a default constructor.
- The access modifier for the default constructor provided by the compiler will be SAME as the access modifier as class. (If the class is public then constructor is also public OR If the class is default then constructor is also default).
- The default constructor given by the compiler will have only '2' access modifiers ie., public & default.
- Compiler will provide a default constructor when there are no constructors in the class.

- We can code/write default constructor (or) parameterized constructor basing upon our programming requirements.
- If we declare a constructor as 'private' then we can restrict the object creation of our class in other classes.
- A Constructor is called simultaneously at the time of object creation by using 'new' keyword.
- In constructor we can write a 'return' statement without returning any value (Just like void method).
- We can create a class object by using "new" keyword and "available constructor" .
- CONSTRUCTOR OVERLOADING IS POSSIBLE OVERRIDDING IS NOT POSSIBLE.

Constructor Vs Method

Constructor	Method
Constructor is used to initialize the state of an object.	Method is used to expose behaviour of an object.
Constructor must not have return type.	Method must have return type.
Constructor is invoked implicitly.	Method is invoked explicitly.
The java compiler provides a default constructor if you don't have any constructor.	Method is not provided by compiler in any case.
Constructor name must be same as the class name.	Method name may or may not be same as class name.

***We need to initialize a class object with the available constructors present in our program otherwise we will be getting a compile time error.

```java
package com.pack1 ;

public class ClassA
{
    void meth1()
    {
        System.out.println("meth1() called");
    }
    public static void main(String[] args)
    {
        ClassA aobj=new ClassA();
        aobj.meth3();
    }

}
```

in this we are not having any construtor if we are not having any construcor the default constructor provided by the compiler

```java
package com.pack1;

public class ClassB
{
    void meth2()
    {
        System.out.println("meth2() called");
    }
    ClassB(String s)
    {
        System.out.println("s : "+s);
    }
    public static void main(String[] args)
    {
        ClassB bobj=new ClassB();
        //bobj.meth3();
    }
}
```

here error because we are having parameterized constructor

```java
package com.pack1 ;

public class ClassA
{
    void meth1()
    {
        System.out.println("meth1() called");
    }
    public static void main(String[] args)
    {
        ClassA aobj=new ClassA();

    }

}
```

```java
package com.pack1;

public class ClassB
{
    void meth2()
    {
        System.out.println("meth2() called");
    }
    ClassB()
    {
        System.out.println("Default constructor");
    }
    ClassB(String s)
    {
        System.out.println("s : "+s);
    }
    public static void main(String[] args)
    {
        ClassB bobj=new ClassB("Java");

    }
}
```

If we declare a constructor as '**private**' then we can restrict the object creation of our class in other classes.

```java
package com.pack1 ;

public class ClassA
{
    void meth1()
    {
        System.out.println(10);
    }
    void meth2()
    {
        System.out.println(20);
    }
    void meth3()
    {
        System.out.println(30);
    }
    private ClassA()
    {

    }
}
```

```java
package com.pack1;

public class ClassB
{
    public static void main(String[] args)
    {
        ClassA aobj1=new ClassA();
        aobj1.meth1();
        aobj1.meth2();
        aobj1.meth3();
    }
}
```

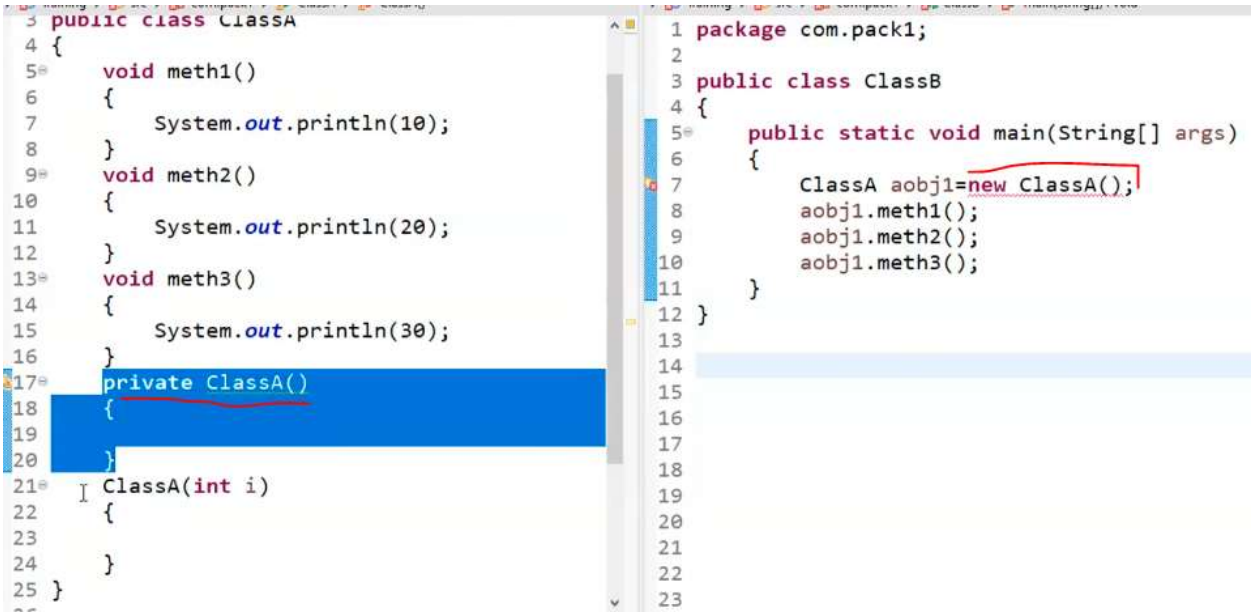
```java
package com.pack1 ;

public class ClassA
{
    void meth1()
    {
        System.out.println(10);
    }
    void meth2()
    {
        System.out.println(20);
    }
    void meth3()
    {
        System.out.println(30);
    }
    public ClassA()
    {

    }
}
```

```java
package com.pack1;

public class ClassB
{
    public static void main(String[] args)
    {
        ClassA aobj1=new ClassA();
        aobj1.meth1();
        aobj1.meth2();
        aobj1.meth3();
    }
}
```



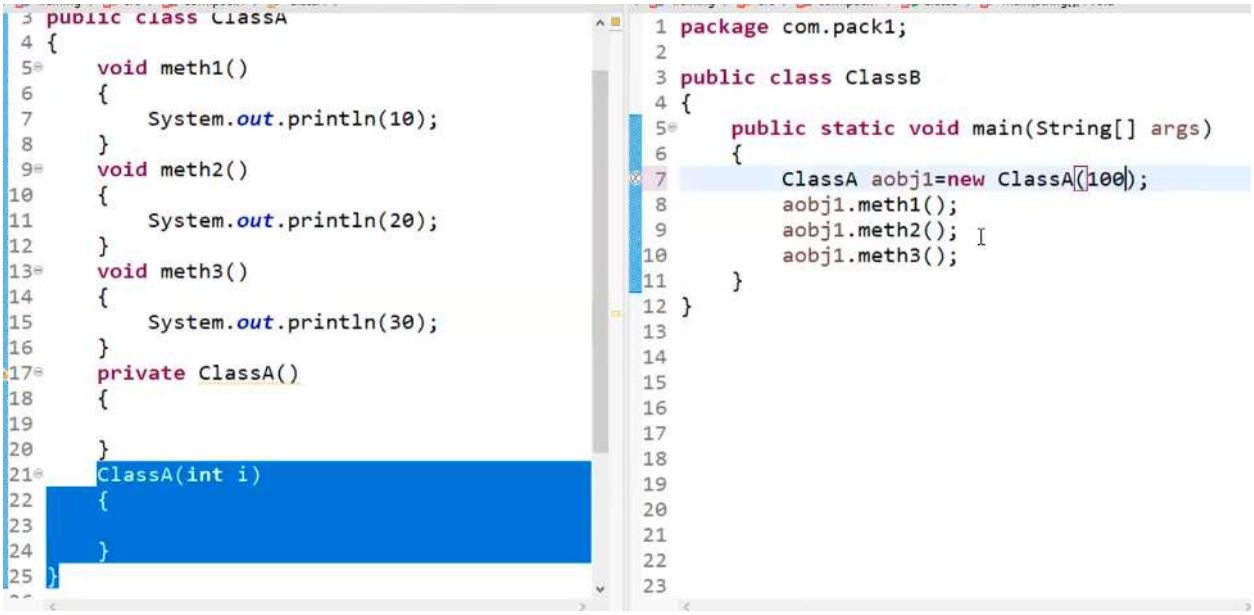
```
public class ClassA
{
    void meth1()
    {
        System.out.println(10);
    }
    void meth2()
    {
        System.out.println(20);
    }
    void meth3()
    {
        System.out.println(30);
    }
    private ClassA()
    {

    }
    ClassA(int i)
    {

    }
}
```

```
package com.pack1;

public class ClassB
{
    public static void main(String[] args)
    {
        ClassA aobj1=new ClassA();
        aobj1.meth1();
        aobj1.meth2();
        aobj1.meth3();
    }
}
```



```
public class ClassA
{
    void meth1()
    {
        System.out.println(10);
    }
    void meth2()
    {
        System.out.println(20);
    }
    void meth3()
    {
        System.out.println(30);
    }
    private ClassA()
    {

    }
    ClassA(int i)
    {

    }
}
```

```
package com.pack1;

public class ClassB
{
    public static void main(String[] args)
    {
        ClassA aobj1=new ClassA(100);
        aobj1.meth1();
        aobj1.meth2();
        aobj1.meth3();
    }
}
```

```java
public class ClassA
{
    void meth1()
    {
        System.out.println(10);
    }
    void meth2()
    {
        System.out.println(20);
    }
    void meth3()
    {
        System.out.println(30);
    }
    private ClassA()
    {

    }
    private ClassA(int i)
    {

    }
}
```

```java
package com.pack1;

public class ClassB
{
    public static void main(String[] args)
    {
        ClassA aobj1=new ClassA(100);
        aobj1.meth1();
        aobj1.meth2();
        aobj1.meth3();
    }
}
```

Just like void methods, inside a constructor also we can write a return statement without returning anything.

```java
package com.pack1 ;

public class ClassA
{
    void meth1()
    {
        return 700;
    }
    ClassA()
    {
        return 800;
    }

}
```

```java
package com.pack1 ;

public class ClassA
{
    void meth1()
    {
        return;
    }
    ClassA()
    {
        return;
    }

}
```

Home work

1) Create a class named 'Programming'. While creating an object of the class, if nothing is passed to it, then the message "I love programming languages" should be printed. If some String is passed to it, then in place of "programming languages" the name of that String variable should be printed.For example, while creating object if we pass "Java", then "I love Java" should be printed.

2)Write a program to print the name of a student by creating a Student class. If no name is passed while creating an object of Student class, then the name should be "Unknown", otherwise the name should be equal to the

String value passed while creating object of Student class.

3) Suppose you have a Bank Account with an initial amount of 500 and you have to add some more amount to it. Create a class 'AddAmount' with a data member named 'amount' with an initial value of 500. Now make two constructors of this class as follows:
1 - without any parameter - no amount will be added then just display your balance with out updating
2 - having a parameter which is the amount that will be added to the account & display the final amount.