

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

**Smart Programming**
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies
City : Mohali (Punjab),
& Chandigarh
(India)

 **WEBSITE :** <http://www.smartprogramming.in>

 **BUY COURSES ON :** <https://courses.smartprogramming.in>

 **YOUTUBE CHANNEL :** Smart Programming (<https://www.youtube.com/c/SmartProgramming>)

 **ANDROID APP :** Smart Programming
(<https://play.google.com/store/apps/details?id=com.smartprogramming>)

 <https://www.facebook.com/smartprogramming.india>

 https://www.instagram.com/smart_programming



Predefined Functional Interfaces

=> Predicate :-

➔ Predicate is a functional interface which is present in java.util.function package

➔ Syntax :-

```
public interface Predicate<T>
{
    boolean test(T t);
    //some default methods are also present
}
```

➔ It improves manageability of the code and also helps in unit-testing

=> Function :-

➔ Function is a functional interface which is present in java.util.function package

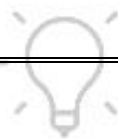
➔ Syntax :

```
public interface Function<T,R>
{
    R apply(T t);
    //some default methods are also present
}
```

**Smart
Programming**

We Educate

We Develop



=> Consumer :-

➔ Consumer is a functional interface which is present in java.util.function package

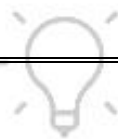
➔ Syntax :

```
public interface Consumer<T>
{
    void accept(T t);
    //default methods..
}
```

**Smart
Programming**

We Educate

We Develop



=> Supplier :-

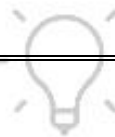
- ➔ Supplier is a functional interface which is present in java.util.function package
- ➔ Syntax :

```
public interface Supplier<R>
{
    R get();
}
```

**Smart
Programming**

We Educate

We Develop



=> What is difference between Predicate, Function, Consumer & Supplier

```
interface Predicate<T>
{
    boolean test(T t);
}
```

It will take an argument and test it and always provide boolean value

```
interface Function<T, R>
{
    R apply(T t);
}
```

It will take an argument and return any value

```
interface Consumer<T>
{
    void accept(T t);
}
```

It will consume any value i.e. it needs an argument but it will not return any value

```
interface Supplier<R>
{
    R get();
}
```

It doesn't need any argument, it will only return the value

Smart Programming

We Educate
We Develop



Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: <https://www.smartprogramming.in/>
<https://courses.smartprogramming.in>

Android App:
<https://play.google.com/store/apps/details?id=com.smartprogramming>

YouTube Channel:
<https://www.youtube.com/c/SmartProgramming>