Blogs

Core java

PageNo: ▼ Go << >>

--Tag-- ▼ Search

Excute Statement before executing main method!

Q: Can you execute a statement before calling main method?
A: Yes, we can define such statements in class static block.   Static block is executed at the time of class loading.

```
public class HelloJava {

   // Static block, it is called when class is loaded.
   static {
      System.out.println("Before main class in static block");
   }

   public static void main(String[] args) {
      System.out.println("Hello Java!");
   }

}
```

Output

```
Before main class in static block
Hello Java!
```

Inheritance

Q Can you inherit String class?
A: No it is final class.

String Program Output : name.toUpperCase()

Q: What will be the output of following program ?

```
String name = “Vijay” ;
name.toUpperCase();
System.out.println(name);
```

Answer:
Output will be

```
Vijay
```

Since String is immutable and cam not be changed, when you call *name.toUpperCase()* it will create a new String "VIJAY" but old string ‘name’ will not be changed.

If you want to print name in upper case then you have to store the reference of new String in a variable then print that. Code will be written like

```
String newName = name.toUpperCase();
System.out.println(newName);
```

Output will be

```
VIJAY
```

Abstract Class

Q: Can you define a Class abstract if it does not have abstract methods?
A: Yes, we can do it by just specifying abstract keyword before class.
Ex. *public abstract class NoAbstractMethod{*

```
public abstract class NoAbstractMethod{
    ...
}
```

Q: Why do you use abstract classes?
A: An abstract class provides partial implementation of functionality.

Abstract classes are used when parent classes need to provide some default behavior (methods) and child classes need to provide some specialized behavior (methods). Application framework use these classes to provide default implementation.

For example, if you have an application framework, an abstract class may provide default services such as event and message handling. Those services allow your application to plug in to your application framework. However, there is some application-specific functionality that only your application can perform.

Q: Why do you use abstract methods?
A: When Child classes need to provide specialized behavior of a method then we define abstract methods.

Q: Can a static method be abstract?A: NO, It is a class method and called with class name so can not be abstract. The abstract keyword cannot be applied to static method declarations. The compiler will reject the class with the error "illegal combination of modifiers".

Q: Can a private method be abstract?
A: NO, Child classes do not have access on private methods so they can not be defined by child classes that is why we can not make them private.

Q: Can I call a static method on an abstract class?A: YES, since an object instance is not required and method is called with Class name.

Define Variables in JAVA Interface

Q: Can you define variable in Interface?
A: Yes, we can define variables in Interface but all variables defined in Interface are by default static and final.

Q: What is the default behavior variable?
A: Default behavior of variable is static and final.

For Ex.

```
public interface Role {
    public int Admin = 1;
    public int Manager = 2;
    public int Guest = 3;
}
```