→ The speciality of finally block is it will be executed always irrespective of whether exception is raised or not raised, and whether handled or not handled.

③ finalize () :–

→ finalize is a method always invoked by Garbage collector Just before destroying an object to perform clean up activities.

→ once finalize method compleates, immediately Galbase Colleetor destroys that Objects.

NOTE:– finally block is responsible to perform clean up activities related to try block. i.e. whatever resources, we opened at the part of try block, will be closed inside finally block.

whereas, finalize () method is responsible to perform clean up activities related to object. i.e. whatever resources associated with the Object will be de-allocated before destroying an Object by using finalize () method.

⇒ various possible combinations of try-catch-finally

→
① try	② try	③ try
{	{	{
}	}	}
catch (X e)	catch (X e)	catch(X e)
{	{	{
} ✓	}	} ✗
	catch (Y e)	catch(x e)
	{	{
	} ✓	}
		i.e. exception X has already been caught.

(4)
```
try
{
}
catch ( x e )
{
}
finally {
}
```
✓

5
```
try
{
}
finally
{
}
```
✓

6
```
try
{
}
catch ( x e )
{
}
try {
}
catch ( y e )
{
}
```
✓

7
```
try
{
}
catch ( x e )
{
}
try
{
}
finally
{
}
```
✓

8
```
try
{
}
```
✗
C.E :—
try without
catch or finally
✗

9
```
catch ( x e )
{
}
```
C.E :— catch without try
✗

10
```
finally
{
}
```
CE :— finally without try
✗

11
```
try
{
}
finally
{
}
catch ( x e )
{
}
```
✗
CE ! — catch without try

12
```
try
{
}
s.o.p ("Hello");
catch ( x e )
{
}
```
✗
CE :— try without catch block
✗

13
```
try
{
}
catch ( x e )
{
}
s.o.p ("Hello");
catch ( y e )
{
}
```
✗
CE :— catch without try

14
```
try
{
}
catch ( x e )
{
}
s.o.p ("Hello")
finally
{
}
```
CE :— finally without try
✗

15
```
try
{
try
{
}
catch ( x e )
{
}
catch ( x e )
{
}
}
```
✓

16
```
try
{
try
{
}
}
catch ( x e )
{
}
```
✗
CE :— try without catch or finally

17
```
try
{
try
{
}
finally
{
}
catch ( x e )
{
}
}
```
✓

18
```
try
{
}
catch ( x e )
{
try
{
}
finally
{
}
}
```
✓

19
```
try
{
}
catch ( x e )
{
}
finally
{
}
```
✗
CE :— finally without try
✗

20
```
try
{
}
catch ( x e )
{
}
finally {
try {
}
catch ( x e )
{
}
}
```
✓

21
```
try
{
}
catch ( x e )
{
}
finally
{
finally
{
}
}
```
✗
CE :— finally without...

22
```
try
{
}
catch ( x e )
{
}
finally {
}
```
✗
CE :— finally without try