Exception handling

EXCEPTION

During the execution of java program, JVM faces abnormal situations based on code declaration
If JVM faces abnormal situation, JVM triggers an event, this event is known as exception.
If exception event get generated then it results in termination of program.
If termination of program takes place, then the code is non-feasible code for execution.
If any event is generated by JVM then programmer need to handle the event so that all the lines present in the program get executed.

EXCEPTION HANDLING

Handling the event generated by JVM during program execution is known as exception handling.
It is recommended to handle events during execution flow.

we can handle exception using 2 ways:
    1: try & catch block
    2: Throw & Throws keywords.

1: handle exception using try & catch block

To handle to handle the event or exception, below blocks are used

SYNTAX
tr…

```
try
{
        //risky code
}
catch(expectedExceptionName objectName)
{
        //empty body
        //handle exception
        //alternate code
}

remaining stat
```

Try block: -

A try block is the block of code (contains a set of statements) in which exceptions can occur; it's used to declare risky code.
The try block is always followed by a catch block.
Multiple try blocks are not allowed.

catch block:

It is use to handled event/exception generated in try block.
Catch block will get executed only if event/exception generated in try block.
catch block should be declared after try block.
Any no of catch block should be declared after try block

```
package ExceptionHandling;
public class sample11
{
        public static void main(String[] args)
        {
                int a=10;
                int b=0;

                int div=0;

                try
                {
                         div=a/b;      //10/0=                risky code
                }
                catch(ArithmeticException e)
                {
                        //empty catch block body
                }

                System.out.println(div);
                System.out.println("Hello GM");
        }
}
```

```
package ExceptionHandling;
public class sample12
{
        public static void main(String[] args)
        {
                String [] ar= {"mahesh","ramesh","suresh"};

                try
                {
                        System.out.println(ar[7]);      //risky code
                }
                catch(ArrayIndexOutOfBoundsException e)
                {
                        //empty catch block body
                }

                System.out.println("Hi");
        }
}
```

```
package ExceptionHandling;
public class sample13
{
        public static void main(String[] args)
        {
                String s1= "abcd";

                try
                {
                        System.out.println(s1.charAt(9));      //risky code
                }
                catch(StringIndexOutOfBoundsException e)
                {
```

```
            //empty catch block body
        }

        System.out.println("Hi");
    }
}


package ExceptionHandling;
public class sample2
{
    public static void main(String[] args)
    {
        String s1= "abcd";

        try
        {
            System.out.println(s1.charAt(8));      //risky code
        }
        catch(StringIndexOutOfBoundsException e)
        {
            System.out.println("StringIndexOutOfBounds Exception handeld");
        }

        System.out.println("Hi");
    }
}


package ExceptionHandling;
public class sample3
{
    public static void main(String[] args)
    {
        String s1= "abcd";

        try
        {
            System.out.println(s1.charAt(8));      //risky code
        }
        catch(StringIndexOutOfBoundsException b)
        {
            System.out.println(s1.charAt(0));              //alternate code
        }

        System.out.println("Hi");
    }
}


package ExceptionHandling;
public class sample4
{
    public static void main(String[] args)
    {
        String s1= "abcd";

        try
        {
            System.out.println(s1.charAt(8));      //risky code
        }
        catch(ArrayIndexOutOfBoundsException b)
        {
            System.out.println("ArrayIndexOutOfBounds Exception handled");
        }
        catch(ArithmeticException b)
        {
            System.out.println("ArithmeticException Exception handled");
        }
        catch(StringIndexOutOfBoundsException b)
        {
```

```
				System.out.println("StringIndexOutOfBounds Exception handled");
			}

			System.out.println("Hi");
		}
}
```

```
package ExceptionHandling;
public class sample5
{
		public static void main(String[] args)
		{
			String s1= "abcd";

			try
			{
				System.out.println(s1.charAt(9));	//risky code
			}
			catch(Exception b)
			{
				b.printStackTrace();
				System.out.println("generic Exception handled");
			}

			System.out.println("Hi");
		}
}
```

```
package ExceptionHandling;
public class sample6
{
		public static void main(String[] args)
		{
			String s1= "abcd";

			try
			{
				System.out.println(s1.charAt(9));	//risky code
			}
			catch (ArrayIndexOutOfBoundsException e)
			{
				System.out.println("ArrayIndexOutOfBounds Exception handled");
			}
			catch (ArithmeticException e)
			{
				System.out.println("Arithmetic Exception handled");
			}
			catch (StringIndexOutOfBoundsException e)
			{
				System.out.println("StringIndexOutOfBounds Exception handled");
			}
			catch(Exception b)
			{
				b.printStackTrace();
				System.out.println("generic Exception handled");
			}

			System.out.println("Hi");
		}
}
```

```
package ExceptionHandling;
public class sample7
{
		public static void main(String[] args)
		{
			String s1= "abcd";
		try
		{
			System.out.println(s1.charAt(9));	//risky code1
		}
		catch(StringIndexOutOfBoundsException e)
		{
			System.out.println("StringIndexOutOfBounds Exception handeld");
		}
		System.out.println("Hi");


		String [] ar= {"mahesh","ramesh","suresh"};
		try
		{
			System.out.println(ar[9]);	//risky code2
		}
		catch(ArrayIndexOutOfBoundsException e)
		{
			System.out.println("ArrayIndexOutOfBounds Exception handled");
		}

		System.out.println("Hello");
	}
}


package ExceptionHandling;
public class sample8
{
	public static void main(String[] args)
	{
		String s1= "abcd";
		int a=10;
		int b=0;


		try		//outer try block
		{
			try		//inner/nested try block
			{
				System.out.println(a/b);		//risky code1
			}
			catch (ArithmeticException e)	//inner catch block
			{
				System.out.println("Arithmetic Exception handled");
			}

			System.out.println(s1.charAt(9));	//risky code2
		}
		catch(StringIndexOutOfBoundsException e)	//outer catch block
		{
			System.out.println("StringIndexOutOfBounds Exception handeld");
		}


		System.out.println("Hi");
	}
}


package ExceptionHandling;
public class sample9
{
	public static void main(String[] args)
	{
		String s1= "abcd";
		try
		{
			System.out.println(s1.charAt(9));	//risky code1
			//data fetch
		}
		catch(StringIndexOutOfBoundsException e)
		{
			System.out.println("StringIndexOutOfBounds Exception handeld");

		}
		finally
		{
			//connection close
			System.out.println("running ");
		}

		System.out.println("Hi");

	}
}
```

```
package ExceptionHandling;
public class sample10
{
	public static void main(String[] args)
	{
		String s1= "abcd";
		try
		{
			System.out.println(s1.charAt(9));     //risky code1
		}
		finally
		{
			System.out.println("running ");
		}


		System.out.println("Hi");
	}
}
```

Finally block

Finally is a block used to close costly resources of current program
Finally block will get executed in all circumstances
Finally block should be followed by catch block. You can declare finally block after try block but it is not recommended
We can declare statements in between catch and finally block but non recommended.

PrintStackTrace():-
It is a method used to get fully qualified details of an exception

Difference between throw and throws:-
Throw is an keyword used to throw new custom exception in current BLC
Throws is an keyword used to show or declare type of exception generate inside method or class

Q1: valid or not

```
		try
		{
			System.out.println(s1.charAt(9));       //risky code
		}
		catch(ArithmeticException e)
		{
			System.out.println("Arithmetic Exception handled");
		}
		catch(ArithmeticException e)
		{
			System.out.println("Arithmetic Exception handled");
		}
		catch(StringIndexOutOfBoundsException a)
		{
			System.out.println("StringIndexOutOfBounds Exception handled");  //failure msg
		}
```

ANS: it will through compile time error

Q2: valid or not

```
		try
		{
			System.out.println(s1.charAt(9));
		}
		catch(Exception e)
		{
			System.out.println("Generic Exception Handled");
			e.printStackTrace();
		}
		catch(ArithmeticException e)
		{
			System.out.println("Arithmetic Exception handled");
		}
		catch(ArrayIndexOutOfBoundsException e)
		{
			System.out.println("ArrayIndexOutOfBounds Exception handled");
		}
```

ANS: it will through compile time error