Unit 5. Exceptions

| 5.1. Introduction to Exceptions  5.2. Exception Types, User defined Exception  5.3. Throw, Throws  5.4. Try, Catch and Finally  5.5. Thread  5.5.1. Introduction to Threads  5.5.2. Thread Model  5.5.3. Priority of Threads  5.5.4. Inter Thread Communication  5.5.5. Synchronization |
| --- |

**5.1. Introduction to Exceptions**

* **Exception :**
* An exception is an unwanted or unexpected event,which occurs during the execution of program.
* It is an object which is thrown at run time,that disrupts the normal flow of the execution.
* Example :

| class test  {  public static void main(String args[ ])  {  System.out.println(1);  System.out.println(2);  System.out.println(3);  System.out.println(4/0);  System.out.println(5);  }  } | **Output ::**  1  2  3  Exception in thread "main" java.lang.ArithmeticException: / by zero |
| --- | --- |

* In above example,**System.out.println(4/0)** generates an exception and due to this exception the next statements of program can not be executed.
* **Exception handling :**
* The exception handling in java is one of the most essential mechanism to handle the runtime errors so that normal flow of the application can be maintained.

**5.2. Exception Types, User defined Exception**

* **Types of Exception :**
* **User defined Exception / Customized Exception / Custom Exception :**

**5.3. Throw, Throws**

* **Throw :**
* throw keyword is used to throw an exception object explicitly.
* throw keyword always present inside the method body.
* throw is followed by instance(object).
* We can throw only one exception at a time.
* **Syntax :**

| **throw** throwable\_instance |
| --- |

* **Throws :**
* Throws keyword is used to declare an exception as well as by pass the caller.
* Throws keyword always used with method signature.
* Throws is followed by class.
* We can handle multiple exception using throws keyword.
* **Syntax :**

| Return\_type method\_name( ) **throws** exception\_class-name\_list  {  //Method code  } |
| --- |

* **Example :**

**5.4. Try, Catch and Finally**

**5.5. Thread**

**5.5.1. Introduction to Threads**

* Thread is a pre-defined class which is available in java-lang package or library.
* Thread is a Basic unit of CPU and it is well known for independent execution.

**5.5.2. Thread Model**

**5.5.3. Priority of Threads :**

* It is possible to assign the priority of thread in java.
* To set this priority java thread class has been provided two predefined methods.

1. setpriority( )
2. getpriority( )

* The thread class has also been provided three predefined final static variable or constant and it's value lies between 1 to 10.

| **Static variable / constant** | **Value** | **Priority** |
| --- | --- | --- |
| Thread.MIN\_PRIORITY | 1 | Minimum |
| Thread.NORM\_PRIORITY | 5 | Default |
| Thread.MAX\_PRIORITY | 10 | Maximum |

* **Example :**

**5.5.4. Inter Thread Communication :**

* When a thread communicates with another thread by sending some signals to ask thread to wait or wakes up another thread from sleep,that is done by inter thread communication.

**5.5.5. Synchronization :**

* Synchronization is a technique through which we can control multiple threads or among the number of threads only one thread will enter inside the**(અધૂરું છે)**
* Synchronization is used to solve data inconsistency problem.
* It is the modifier applicable only for methods and blocks.
* It can not apply for classes and variables.
* Synchronized keyword in java creates a block of code known as critical section.
* To enter a critical section,a thread needs to obtain the corresponding object's lock.
* **Syntax :**

| Synchronized (object)  {  //Statements  } |
| --- |