**Array Objects**

**Array syntax**

**datatype arrayName[]=new DataType[size];**

**int abc[]=new int[10];**

**Product p1 =new Product();**

**Product products[]=new Product[10];**

**1 , 0 or 10 of product object.**

**products[0]=new Product();**

**products[1]=new Product();**

**products[2]=new Product();**

**service class : this class contains pure business methods.**

**Pre defined packages or built in package**

**Java provided 2 main root package**

**java javax root package**

**lang sql**

**io net**

**util servlet**

**sql ejb**

**net jms**

**etc etc**

**by default every java program imported lang package.**

**lang package classes and interfaces**

**Exception and type of exception classes**

**Thread and Runnable interface**

**String**

**StringBuffer**

**StringBuilder**

**Object**

**Class**

**Cloneable**

**System**

**Runtime**

**Etc**

**Exception Handling**

**Exception is an object or memory which occurs when unexcepted or abnormal condition occurs during the execution of program.**

**Java Program**

**Compile program Run program**

**javac (java compiler) java (java interpreter)**

**compile time error run time error**

**syntax error**

**or typo error**

**Run time error**

**Error Exception**

**Error : Error is a pre defined class in java. Part of lang package. The error which generate at run time which we can’t handle it. JVM Crash, software or hardware issue etc.**

**Exception : Error is a pre defined class in java. Part of lang package. The exception is a type of error which generate at run time which we can handle it. Divided by 0, wrong array index position.**

**Exception**

**Checked exception un checked exception**

**RuntimeException**

**IOException ArithmeticException**

**FileNotFoundException ArrayIndexOutOfBoundsException**

**SQLException NumberFormatException**

**NullPointerException**

**Etc**

**To handle checked as well as un checked exception in java. Java provided 5 keywords.**

1. **try**
2. **catch**
3. **finally**
4. **throw**
5. **throws**

**try catch block**

**syntax**

**try{**

**}catch(Exception e) {**

**}**

**Try with multiple catch block**

**try {**

**}catch(ArithmeticException e){**

**}catch(ArrayIndexOutOfBoundsExcepton e) {**

**}catch(NumberFormatException e) {**

**}**

**Finally block**

**Finally is a block which execute 100% sure if exception generate or not.**

**Try block 🡪 the code which generate exception those code we need to keep in try block.**

**Catch block -🡪 this block execute only if any exception generate. No exception no catch block.**

**try**

**catch catch catch catch finally**

**catch finally catch**

**catch finally**

**Try{**

**Open the file**

**Read or write or copy**

**}catch(Exception e) {**

**}finally {**

**Close the resources**

**}**

**Connect database**

**Store or retrieve**

**Close the connection**

**throw : throw keyword is use to generate or raise pre defined or user defined exception base upon conditions.**