Java – Exceptions

An exception (or exceptional event) is a problem that arises during the execution of a program. When an **Exception** occurs the normal flow of the program is disrupted and the program/Application terminates abnormally, which is not recommended, therefore, these exceptions are to be handled

Catching Exceptions

A method catches an exception using a combination of the **try** and **catch**keywords.

```
try {
   // Protected code
} catch (ExceptionName e1) {
   // Catch block
}
```

```
  public static void main(String args[]) {

      try {

         int a[] = new int[2];

         System.out.println("Access element three :" + a[3]);

      } catch (ArrayIndexOutOfBoundsException e) {

         System.out.println("Exception thrown  :" + e);

      }

      System.out.println("Out of the block");

   }
```

Multiple Catch Blocks

```
try {
   // Protected code
```

```
} catch (ExceptionType1 e1) {
   // Catch block
} catch (ExceptionType2 e2) {
   // Catch block
} catch (ExceptionType3 e3) {
   // Catch block
}
```

Example:

```
ublic class InsufficientFundsException extends Exception {

   private double amount;


   public InsufficientFundsException(double amount) {

      this.amount = amount;

   }


   public double getAmount() {

      return amount;

   }

}
```

Nxt

```
public class CheckingAccount {

   private double balance;

   private int number;


   public CheckingAccount(int number) {

      this.number = number;

   }
```

```
   public void deposit(double amount) {
      balance += amount;
   }

   public void withdraw(double amount) throws InsufficientFundsException {
      if(amount <= balance) {
         balance -= amount;
      }else {
         double needs = amount - balance;
         throw new InsufficientFundsException(needs);
      }
   }

   public double getBalance() {
      return balance;
   }

   public int getNumber() {
      return number;
   }
}
public class BankDemo {

   public static void main(String [] args) {
      CheckingAccount c = new CheckingAccount(101);
      System.out.println("Depositing $500...");
      c.deposit(500.00);
```

```
      try {
         System.out.println("\nWithdrawing $100...");
         c.withdraw(100.00);
         System.out.println("\nWithdrawing $600...");
         c.withdraw(600.00);
      } catch (InsufficientFundsException e) {
         System.out.println("Sorry, but you are short $" + e.getAmount());
         e.printStackTrace();
      }
   }
}
```

```
Depositing $500...

Withdrawing $100...

Withdrawing $600...
Sorry, but you are short $200.0
InsufficientFundsException
        at CheckingAccount.withdraw(CheckingAccount.java:25)
        at BankDemo.main(BankDemo.java:13)
```