A New Mathematical Model of Exception Handling in Java Object Oriented Language

Manoj Kumar Srivastav
Indira Gandhi National Open University
St. Xavier's College (Autonomous), Kolkata, Pin Code-700016,
India

Asoke Nath, Ph.D
Associate Professor, Department of Computer Science, St. Xavier's College (Autonomous),Kolkata, Pin code-700016, India

ABSTRACT

Srivastav et al already published several papers where the authors tried to explain the various aspects of object oriented language like Java using some simple mathematical model or functions. The authors have already studied the general properties of Java language such as object, class, inheritance, multithreaded concepts using simple mathematical concepts. In the present paper the authors have tried to explore the concept of exception handling in java language using discontinuity in a function. The exceptions are normally come during runtime and some of the exceptions are repairable and some may not be repairable. In the present paper the authors tried to explain both repairable exception and also irreparable exceptions by considering simple continuity and discontinuity in functions.

Keywords

Exception handling, discontinuity ,function, continuity, object

1. INTRODUCTION

Compilation is a process that translates a program in one language (the source language) into an equivalent program in another language (the object or target language). An important role of any compiler is the detection and reporting of errors. The source code is a high-level programming language which may be C or C++ or C# or Java or any other language. The target code may be machine language code or assembly code. Thus compilation is a fundamental concept in the production of software: it is the link between the (abstract) world of application development and the low-level world of application execution on machines. Functions are the major tools for describing the real world in mathematical terms. A function plays a very important role in the programming language like C, Java etc. The set of all possible input values is called the domain of the function. The set of all output values is called the range of the function. In calculus , Leohard Euler (1707-1783) invented a symbolic way to say "y is function of x" i.e. y=f(x) (" y equals f of x").In this notation, the symbol 'f ' represents the function. The letter x, called the independent variable, represent an input value from the domain of f, and y, the dependent variable, represent the corresponding output values f(x) in the range of f.

The authors have found a close relationship between mathematical functions and exceptional handling. There are two types of mathematical function :(i)Continuous function and (ii)Discontinuous function. The source function in a programming language may be continuous or Discontinuous function . When the source function in the programming language, either in procedural language like C or in object oriented language like Java, is continuous there is no interruption in the flow of the program but on the other hand if the source function is discontinuous function , there will be an interruption in the programs that is called runtime error. In procedural language there is no way to prevent run time errors as there is no concept called exception handling. The advantage of object oriented programming language like Java is that one can trap the runtime errors using exception handling routines which is not there in so called procedural language like "C".

2. MATHEMATICAL FUNCTION THAT ARE GENERALLY TAKEN AS SOURCE FUNCTION/ VALUE

Real –Valued Functions : A function $f : X \longrightarrow Y$ is a correspondence(or a rule) that assigns to each x ε X, a unique elements y=f(x) in Y. X is called the domain of f and Y, the co-domain of f. The unique element y=f(x) in Y, that is assigned to x, by the function f is called the value of f at x. The symbol x, that take the values from the dom f, is called the independent variable, while the symbol y to denote f(x), is called the dependent variable.

The subset f(X)={y | y=f(x), x ε X } is subset of Y , is called the range of f.

If Y ⊆ R then the function $f : X \longrightarrow Y$, takes up real value only .For this reason, it is called a real-valued function or simply a real function.

If X ⊆ R , then the independent variable also takes up real value only. In this case , the function $f : X \longrightarrow Y$ is called a real function of real variable.

Some Elementary Real Function:

(i)A constant function f_c :$f_c(x)$= c for all x ε R dom f_c=R, $f_c(R)$={c}

(ii)The identity function, I:I(x)=x, for all x ε R, dom I=R, I(R)=R.

(iii)Reciprocal function f :f(x)=1/x, for all x ε (R-{0}), dom f=R-{0}, f(R-{0})=R

(iv)Absolute value function'| |' : |x|=x, if x> 0 and |x|=-x if x<0.

(v)Positive integral Power function

,x^n: $x \longrightarrow x^n$ x ε R , n a positive integer. dom x^n=R, ran x^n = R, if n is odd And ran x^n= R^+ ∪ {0}, if n is even.

(vi)Polynomial Function P_n: $P_n(x)= a_0x^n + a_1x^{n-1}+ a_2x^{n-2}+\ldots\ldots+a_n$, Where $a_0, a_1, a_2, \ldots\ldots\ldots, a_n$ are constants, $a_0 \neq 0$; n a positive integer .dom P_n =R, ran $P_n \subseteq$ R

(vii)Rational function , R :R(x)=P(x)/Q(x), x ε R Q(x)≠ 0, P(x), Q(x) are polynomial.dom 'R'=R-{x | Q(x) =0} , ran 'R' ⊆ R (viii)Bracket function, [.]:[x]=greatest integer not greater than x, x ε R

dom'[ ]'=R, ran'[ ]'=Z

(ix)qth root function, $x^{1/q\,:}$ y= $x^{1/q}$ iff y^q=x, q is a positive integer.

dom'$x^{1/q}$ ' =R^+ {0}, if q is even

dom'$x^{1/q}$ =R , if q is odd. ran '$x^{1/q'}$ ⊆ R.

(x)The sin function, sin : dom sine =R ran sine=[-1,1].

sin(-x)= -sin x for all x ε R

sin(2 π +x)=sin x for all x ε R

sin(A+B)= sinA cos B+ cos A sin B, for all A, B ε R

(xi)The cosine function, cos : dom cosine =R, for cosine=[-1,1].

cos(-x)= cos x for all x ε R

cos(2 π +x)=cos x for all x ε R

cos(A+B)= cosA cos B-sin A sin B, for all A, B ε R

(xii) Exponential function , $e^{x\,:}$

e^x =1+x +x^2/2!+…….+ x^n/n! +……..

dom e^x =R, ran e^x =(0, ∞)

e^{x+y} = e^x .e^y and e^{-x} =1/$e^{-x,}$ x,y ε R

(xiii) Logarithm function , log x: y=log x iff e^y=x, x>0

dom log x= $R^{+,}$ ran log x=R

log(xy)=log x+log y, log (1/x)=-log x, x,y ε R^+

(xiv) Inverse sine (or arc sine), $\sin^{-1}$:

y= $\sin^{-1}$x iff x=siny

dom $\sin^{-1}$=[-1,1], ran $\sin^{-1}$=[- π /2, cπ /2]

(xv) Inverse cosine (or arc cosine function), cos $^{-1}$:

y= cos $^{-1}$x iff x=cosy

dom $\cos^{-1}$=[-1,1], ran $\cos^{-1-}$=[0, π]

3. MATHEMATICAL DESCRIPTION OF THE EXCEPTIONAL HANDLING IN JAVA

The **exception handling in java** is one of the powerful mechanism to handle the runtime errors so that normal flow of the application can be maintained. **Dictionary Meaning of** Exception is an abnormal condition .In java, exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime. Similarly, a function f(x) is commonly said to be continuous provided its graph is a continuous curve, and if there is any discontinuity or break at any point on the curve, the function is said to be discontinuous for the corresponding value of x. The general notations of continuity of a function f(x) for any value of the variable x requires that the function should be finite at that point, and for very small change in x, the change in the value of f(x) should be small, or in other words as we approach the particular value of x from either side the function should also approach the corresponding value of f(x), and ultimately coincide with it at that point. If f(x) be non-existent at a point, so the corresponding point on the graph is missing, or else, if the value of f(x) suddenly jumps as x passes from one side to the other or the particular value, or f(x) becomes infinitely large at a point, then the function is discontinuous there. In the object- oriented programming language, if we define a mapping f:(X, **τ**) → (Y, σ) such that f(address of object)=address of new object [Here X and Y are object set and are **τ** set of sets with respect to object X and σ are set of sets with respect to object Y .Actually represents classes for domain and represents class for range in the above function and f represents main function.] then for each open subset V of Y ,the set f^{-1}(V)is open subset of X. Since Exception is an object which is thrown at run time . So the concept of continuity and discontinuity may be apply only on the the object i.e., simply on the sets only.

Intuitively, a function is **continuous** at *x* = *c* if we can draw it without lifting our pen from the paper. Formally, a function is continuous at x = c on the open interval (a, b) if the following three conditions are met :

(i)f(c) is defined
(ii)lim f (x) exists
x→c
(iii) lim f (x) =f (c)
x → *c*

If f be a function defined in an interval I and x_0 ε I, f is said to be **discontinuous** at $x_{0,}$ if it is not continuous at $x_{0.}$ Also in that case , the point x_0 is said to be a power of discontinuity or simply a discontinuity of f.

(a) Types of Discontiuity:
(i)Removable Discontinuity : Let f be a function defined on an interval I and p ε I. If

lim f(x) exists but not equal f(p), then p is called

x → p

a removable discontinuity of f .

(ii)Discontinuity of first kind: Let f be a function defined on an interval I and p ε I. If

f(p-0) and f(p+0) both exists but are unequal, then p is called discontinuity of the first kind.

If f(p)= f(p+0), but f(p-0) ≠ f(p), then p is called a discontinuity of first kind from the left.

If f(p-0)= f(p), but f(p) ≠ f(p+0), then p is called a discontinuity of first kind from the right.

(iii)Discontinuity of second kind : Let f be a function defined on an interval I and p ε I. If neither f(p-0) and f(p+0) exists , then p is called a discontinuity of the second kind of f

If one of the two one-sided limits of f, f(p-0) or f(p+0) exists while the other does not, then p is called a point of mixed discontinuity or discontinuity of f of second kind, from the left or the right as the case may be.

(b)Types of Exception in object Oriented Programming(in Java)
There are mainly two types of exceptions: checked and unchecked . The exception error considered as unchecked exception. The sun micro system says there are three types of exceptions:

(i)Checked Exception

(ii)Unchecked Exception

(iii)Error

(i) Checked Exception
The classes that extend Throwable class except Runtime Exception and Error are known as checked exceptions e.g.IOException, SQLException etc. Checked exceptions are checked at compile-time.

(ii)Unchecked Exception
The classes that extend Runtime Exception are known as unchecked exceptions e.g. Arithmetic Exception, NullPointerException, ArrayIndexOutOfBoundsException etc. Unchecked exceptions are not checked at compile-time rather they are checked at runtime.

(iii)Error
Error is irrecoverable e.g. OutOfMemoryError, VirtualMachineError, AssertionError etc.

The purpose of Exceptional handling mechanism is to provide a means to detect and report an “exceptional circumstance” so that appropriate action be taken. The mechanism suggests incorporation of separate error handling code that performs the following tasks:

(a)Find the problem (Hit the Exception).

(b)Inform that an error has occurred(Throw the exception)

(c)Receive the error information(Catch the exception)

(d)Take corrective actions(Handle the exception)

The error handling code basically consists of two segments, one to detect errors and to throw exceptions and the other to catch exception and to take appropriate actions. Similarly Removal discontinuity is also a discontinuous function .But it can be made to continuous at the discontinuous points by redefining its point at that point .Exceptional handling is also treating that function /value which are undefined. Therefore there is need of redefining the function using some special coding in exceptional handling .

4. SOME EXAMPLE OF FUNCTION WHERE REMOVAL DISCONTINUITY OCCURS ON DISCONTINOUS FUNCTION

When a graph is drawn for a function, there may exist hole in a graph .i.e if there exist a hole in the graph, the given function is discontinuous .Therefore, discontinuity of the function can be repaired by filling a single point In other words, intituively, removable discontinuity is a point at which a graph is not connected but can be made connected by filling in a single point.Formally, a removable discontinuity is one at which the limit of the function exists but does not equal the value of the function at that point; this may be because the function does not exist at that point.

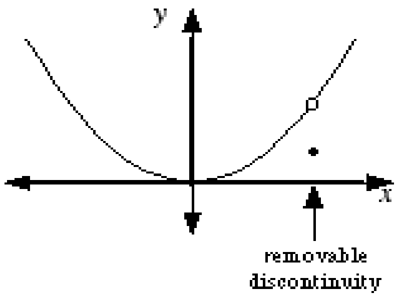


Figure -1 For removable discontinuity

The set of points of removable discontinuities of a function is countable.

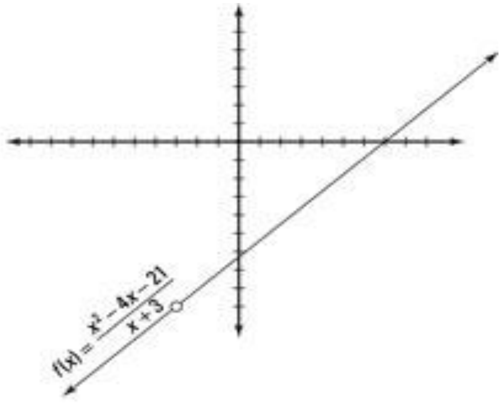
Example 1: If the function factors and the bottom term cancels, the discontinuity at the *x*-value for which the denominator was zero is removable, so the graph has a hole in it.

Consider , the function f : X $\longrightarrow$ R

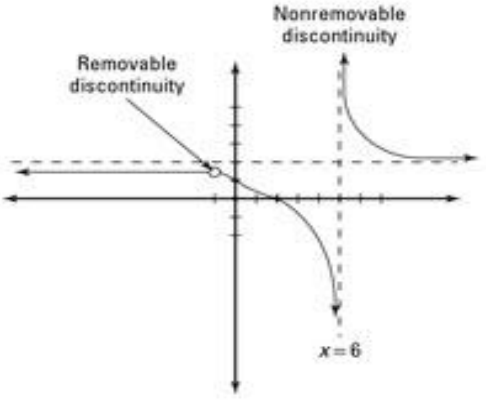
$f(x)=(x^2-4x-21)/(x+3)$

= ((x+3) *(x-7)) / (x+3) for x belongs to set X and here R is representing set of real numbers.

After canceling, it leaves with *x* – 7. Therefore *x* + 3 = 0 (or *x* = –3) is a removable discontinuity — the graph has a hole, as shown in figure a.



a.



b.

Example 2 :If a term doesn't cancel, the discontinuity at this *x* value corresponding to this term for which the denominator is zero is non -removable, and the graph has a vertical asymptote.

Consider a function g: X$\longrightarrow$ R

The following function factors as shown:

$g(x) = (x^2 - x-2)/(x^2-5x-6)$

$=((x+1)*(x-2))/((x+1)*(x-6))$

for x belongs to set X and here R is representing set of real number. Because the *x* + 1 cancels, there is a removable discontinuity at *x* = –1 (i.e, there is a hole in the graph , not an asymptote). But the *x* – 6 didn't cancel in the denominator, so, there is a nonremovable discontinuity at *x* = 6. This discontinuity creates a vertical asymptote in the graph at *x* = 6. The figure b above shows the graph of *g*(*x*)

5. SOME EXAMPLE OF FUNCTION WHERE EXCEPTION OCCURS ON SOURCE FUNCTION

Common scenarios where exceptions may occur

There are given some scenarios where unchecked exceptions can occur. They are as follows:

(i) Scenario where ArithmeticException occurs

If any number is divided by zero, there occurs an ArithmeticException.

```
int a=50/0;//ArithmeticException

   //exception_1.java: Write a program to test number format

    //exception

    import java.io.*;

    class exception_1

    {

        public static void main(String args[])throws IOException

        {

  BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        int i,nvalid,ninvalid,num,a=5,b=6,c=6,d;

        nvalid=ninvalid=0;

            for(i=0;i<args.length;i++)

            {

                try

                {

                num=Integer.parseInt(args[i]);

                d=a/(b-c);

                nvalid=nvalid+1;

                System.out.println("\nValid Number="+num);

                }

                catch(ArithmeticException e)

                {

                System.out.println("\nDivide by zero\n");

                }


                catch(NumberFormatException e)

                {

                ninvalid=ninvalid+1;

                System.out.println("\nInvalid number="+args[i]);

                }

            }

        System.out.println("\nTotal Valid numbers="+nvalid+

        "Total Invalid numbers="+ninvalid);

        }

    }
```

Exceptional Handling	Removable Discontinuity/ condition

a=5,b=6,c=6
d=a/(b-c);
ArithmeticException e
	Consider a function f(x)=5/(6-x)
	This function is continuous at all points of real numbers except at x=6,
	So for this function e removable condition may be taken as
	f(x)=5/(6-x) =c ,a constant value at x=6

(ii)Scenario where NumberFormatException occurs

The wrong formatting of any value, may occur NumberFormatException. Suppose there is a string variable that has characters, converting this variable into digit will occur

NumberFormatException.

```
   String s="abc";

int i=Integer.parseInt(s);//NumberFormatException

 //except_1.java : program to check number format

    import java.io.*;

    class except_1

    {

        public static void main(String args[])throws IOException

        {

        int inv_count,vld_count, i;

        int n,number;

        n=args.length;

        inv_count=vld_count=0;

            for(i=0;i<n;i++)

            {

                try

                {

                number=Integer.parseInt(args[i]);

                }

                catch(NumberFormatException e)

                {

                inv_count++;

                System.out.println("Invalid number="+args[i]);
                continue;

                }

            System.out.println("Valid number="+args[i]);

            vld_count++;

            }

        System.out.println("Valid numbers="+vld_count);

        System.out.println("Invalid numbers="+inv_count);

        }

    }
```

Exceptional Handling	Removable Discontinuity/condition
The wrong formatting of any value, may occur NumberFormatException. Suppose there is a string variable that have characters, converting this variable into digit will occur NumberFormatException. String s="abc"; int i=Integer.parseInt(s);//NumberFormatException	Since string are linear combination of character Each character possesses some ASCII value . Consider a function $f(x)=f_1(x)+ f_2(x)+ f_3(x)+……..+ f_n(x)$.Suppose some functions have integer value and some have float value .Then there may be problem to find the output as integer value. Consider a function $f_i(x)=x_i.x_d$ is equal to $f_i(x)=x_i$ for i=1,2,3,……n. x_i represents integer parts and x_d decimal parts.

(iii) Scenario where ArrayIndexOutOfBoundsException occurs

The ArrayIndexOutOfBoundsException will arise if the index of the array is taken some invalid value such as negative or may be more than the upper limit of the array. An example is given below:

```
    int a[]=new int[5];
a[10]=50; //ArrayIndexOutOfBoundsException
```

//xxexception_3.java: Write a program to use ArrayStoreExecption to implement when wrong data type is introduced in an array

```
    import java.io.*;

    class xxexception_3

    {

        public static void main(String args[])throws IOException

        {

        int a[]=new int[5];

        int i,n,s,m;

    BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        System.out.print("\nEnter size of array(1-5) :");

        n=Integer.parseInt(br.readLine());

        s=0;

        m=0;

            for(i=0;i<n;i++)

            {

                try

                {

                System.out.print("a["+i+"]=");

                a[i]=Integer.parseInt(br.readLine());

                m++;

                s=s+a[i];

                }

                catch(NumberFormatException e)

                {

            System.out.println("\nEntered a wrong data type\n");

                }

              catch(ArrayIndexOutOfBoundsException e)

                {

          System.out.println("\nIndex of Array is out of Range\n");

                }

            }

        System.out.println("Sum of "+m+" numbers="+s);

       }

    }
```

Exceptional Handling	Mathematical Description

If the index of an array is invalid then it would result an error ArrayIndexOutOfBoundsException as shown below:
int a[]=new int[5];
a[10]=50; //ArrayIndexOutOfBoundsException
ArrayIndexOutOfBoundsException e)
	Consider a function f(x)=sin x for x ε [-π/2, π/2] .This function is continuous in this interval. Suppose if someone wants to find the value at x=π , then this value is not lies in domain [-π/2, π/2] of f(x).

However, it is not possible to change all the discontinuous functions to continuous functions by removing some condition. Similarly, it is not possible to implement all possible exceptions in object oriented programming language In the table below a comparative is made among exceptional handling, cause of exceptions and the mathematical descriptions

.Exceptional Handling	Cause of Exception	Mathematical Description
FileNotFoundException	Caused by an attempt to access a non- existent file	Files are stream of characters. Characters makes one to one correspondence with the ASCII value. ASCII values are subset of real numbers. Consider a function like func(x)=1/x , the func(x) does not exist at x=0,so to overcome this difficulty the following condition may be taken: f(x)=1/x, x≠0 =0, x=0.
IOException	Caused by general I/O failures ,such as inability to read from a file. These are the signals that an I/O exception of some sort has occurred	In the some composite function (like product/division) is made in such way that individual is continuity is not considered. Let $f(x)=x^2\sin(1/x)$, x≠0 =0, x=0. is a continuous function where individual discontinuity is not taken.
OutofMemoryException	Caused when there is not enough memory to allocate a new object.	Let I=[a,b] be a closed and bounded interval and let function $f : I \longrightarrow R$ be continuous on I. Then f is bounded on I.Also there is a point cε[a,b] such that f(c)=supf(x) xεI and also there is a point dε[a,b] such that f(d)=inf f(x) xεI
StackOverFlowException	Caused when the system runs out of stack space	Let a function $g : R \longrightarrow R$ is unbounded on every neighbourhood of a point cεR. In that case g is discontinuous at c whether g

		is defined or not.
StringIndexOutofBoundsException	Caused when a program attempts to access a nonexistent character position a string.	This type of case is repairable. There is not any impact on the function even if there is not one-one to mapping between index set and functional value.

(iv)Handling of some exception with mathematical explanation

Exception syntax	**Mathematical Explanation**

Syntax of try with catch block
try{
...
}catch(Exception_class_Name reference){}
	If $\lim_{x\to a} f(x) = L$ but f(a) is not defined
	then the discontinuity at x=a can be removed by defining f(a)=L

try{
...
}finally{}
	When a finally block is defined, this is guaranteed to execute, regardless of whether or not in exception is thrown .There is little change in the mathematics ,which states that " if f is uniformly continuous on an interval I then it is necessarily continuous on I whereas converse may not be true."

Multiple catch block:
If we have to perform different tasks at the occurrence
Of different Exceptions, use multiple catch block.
Syntax: try{
...
}catch(Exception_class1_Name reference) {}
{
Statement
} catch(Exception_class-n_Name reference){}
	If it is possible to remove the discontinuity at each discontinuous point we have to remove the discontinuity at each point respectively one by one

Nested try block:
Syntax:
....
try
{
statement 1;
statement 2;
try
{
statement 1;
statement 2;
}
catch(Exception e)
{
}
}
catch(Exception e)
{
}
....
	We know that composition of a continuous function is continuous function .so to check the continuity of function f1(f2(f3…..(fn))) , it is required to check the continuity of f n first and if it is continuous then o.k. otherwise it is required to remove its discontinuity .Next we have to check the continuity of f(n-1) and apply the same process and so on and lastly we check the continuity of f1.

6. CONCLUSION AND FUTURE SCOPE

The core advantage of exception handling is to maintain the normal flow of the application. Exception normally disrupts the normal flow of the application that is why se exception handling use. The future scope of Exceptional Handling lies that one can find different method /coding to handle different discontinuous functions. It can be used to evaluate integration, Riemann integration, It is known that every continuous function is integrable but converse may not be true. Riemann Integral also deals with discontinuous functions. There is lots of scope to find different methodologies to get the output of different integrable function .It is still believed that the graph of the function could be traced with a" free motion of hand", i.e function were still believed to be piecewise smooth. So, there is scope to think from continuous and discontinuous function for development of exceptional handling in java.

7. REFERENCES

[1] Manoj Kumar Srivastav , Asoke Nath Mathematical modeling of various statements of C-type Language, , International Journal of Advanced Computer Research(IJACR), Vol-3,Number-1, Issue-13, Page:79-87 Dec(2013).

[2] Manoj Kumar Srivastava, Asoke Nath Mathematical Description of variables, pointers, structures, Unions used in C-type language, , Joiurnal of Global Research Computer Science, Vol-5, No-2, Page:24-29, Feb(2014)

[3] Manoj Kumar Srivastav, Asoke Nath, A Mathematical Modeling of Object Oriented Programming Language : a case study on Java programming language, Current Trends in Technology and Science(CTTS) Vol-3, Issue-3, Page 134-141,(2014).

[4] Manoj Kumar Srivastav, Asoke Nath, Analysis of Compilation Errors, Runtime Errors, Reliability and Validity of a Program: A Case Study on C-language : IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 3, May 2014. Page-281-288.

[5] Mathematical foundations of Multithreaded programming concepts in Java language,Manoj Kumar Srivastav, Dr. Asoke Nath, International Journal of Innovative Research in Information Security(IJIRIS)Volume1 Issue4(October2014)Page 6-17

[6] E Balaguruswamy. 2011- Programming with Java-TataMcGrawHill EductionPrivate Limited.

[7] Sachin Malhotra, Sourabh Choudhary.2012 Programming in Java- OXFORD University PressHerbert Schildt,2001 Java TM 2: The Complete Reference, TataMcGraw-Hill Publishing Company Limited,

[8] Satish Jain,Vineeta Pillai,Kratika.2011 Introduction to Object Oriented Programming through Java,BPB Publications

[9] S.K.Mapa, 1998 Real Analysis , Asoke Prakasan

[10] S.K.Mapa,2000 Higher Algebra, -Sarat Book Distribution,

[11] K.D.Joshi-Topology.New Age International Publiocation.

[12] J.N Sharma.2000 "Topology", Krishna Prakashan Media(p)Ltd., Meerut,

[13] R.S.aggarwal.2000,Topology-S.chand publication.

[14] James R Munkers. 2001,Topology- Pearson Education Asia,

[15] www.javatpoint.com

[16] R.D.Sharma, 2011, Mathematics class xi-. Danapat Rai Publications(P) LTD.

[17] Inder K. Rana. 1999 An Introduction to measure and integration- Narosa publication House

[18] Tom M. Apostal. 1997 Mathematical Analysis- Narosa Publishing House

[19] http://www.mathwords.com/r/removable_discontinuity.htm

[20] http://hungry4java.blogspot.in/2014_07_01_archive.html

[21] S.C.Malik,SavitaArora .2006 Mathematical Analysis", New Age International(P) Limited Publishers