

[Sekiranya kamu mengendalikan pelbagai pengecualian di dalam blok try yang sama, kamu mesti mengendalikan kelas pengecualian yang khusus sebelum jelas pengecualian yang umum.]

1. Consider the code structure below and answer the questions that follow.  
[Perhatikan struktur berikut dan jawab soalan-soalan yang dikemukakan.]
- ```

try
    {... code0...}
    catch (Exception1 e1)
        {... code1...}
    catch (Exception2 e2)
        {... code2...}
    finally
        {... code3...}
    code4...

```
- i. Which lines (code0, code1, code2, code3, code4) execute if no exception is thrown?  
[Baris mana (code0, code1, code2, code3, code4) yang akan dilaksanakan jika tiada exception yang dibuang?]
  - ii. Which lines (code0, code1, code2, code3, code4) execute if an exception of type Exception1 is thrown in code0?  
[Baris mana (code0, code1, code2, code3, code4) yang akan dilaksanakan jika exception berjenis Exception1 dibuang dalam code0?]
  - iii. Which lines (code0, code1, code2, code3, code4) execute if an exception of type Exception 2 is thrown in code0?  
[Baris mana (code0, code1, code2, code3, code4) yang akan dilaksanakan jika exception berjenis Exception2 dibuang dalam code0?]
  - iv. Which lines (code0, code1, code2, code3, code4) execute if an exception is thrown in code0 that belongs to neither Exception1 nor Exception2?  
[Baris mana (code0, code1, code2, code3, code4) yang akan dilaksanakan pada code0 yang tidak dimiliki oleh Exception1 atau pun Exception2?]
2. Compile and run Program 10.4 using data in Figure 10.3. Understand what each line of the statement in the program does. In your own words, explain why such output is printed.  
[Kompil dan laksanakan Aturcara 10.4 dengan menggunakan data di Rajah 10.3. Fahamkan apa pernyataan pada setiap baris di dalam aturcara tersebut lakukan. Di dalam ayat kamu sendiri, terangkan kenapa hasil tersebut dicetak.]

```

1 //Program 10.4
2 import java.util.*;
3 import java.io.*;
4
5 public class TestSumException
6 {
7     public static void main (String[]a)
8     {
9         double num[] = new double[5];
10        double total=0;
11        Scanner in = new Scanner(System.in);
12        try
13        {
14            System.out.print("How many floating point
15 numbers do you
16 want to input?");
17            int max= in.nextInt();
18
19            System.out.println("Enter "+max+" numbers");
20
21            for(int i=0;i<max;i++)
22            {
23                num[i]=in.nextDouble();
24                total+=num[i];
25            }
26            System.out.println("End of Program");
27            System.out.println("Total: "+total);
28        }
29        catch(InputMismatchException e)
30        {
31            System.out.println("You input a wrong data
32 type");
33        }
34        catch(IndexOutOfBoundsException e)
35        {
36            System.out.println("Index exceeds the array
37 size");
38        }
39        finally
40        {
41            System.out.println("Good Day");
42        }
43    }
44 }

```

3. Compile and run Program 10.5. Understand what each line of the statement in the program does. In your own words, explain **why such output is printed**.

[*Kompil dan laksanakan Aturcara 10.5. Fahamkan apa penyataan pada setiap baris di dalam aturcara tersebut lakukan. Di dalam ayat kamu sendiri, terangkan kenapa hasil tersebut dicetak.*]

```

1 //Program 10.5
2 public class ExceptionTest
3 {
4     public static void main(String[]a)
5     {

```

```

6      int number;
7      String str;
8
9      try
10     {
11         str="xyz";
12         number=Integer.parseInt(str);
13         System.out.println("A");
14     }
15
16     catch(NumberFormatException e)
17     {
18         System.out.println("B");
19     }
20
21     catch(IllegalArgumentException e)
22     {
23         System.out.println("C");
24     }
25
26     System.out.println("D");
27 }
28 }
```

4. Which exception object may be thrown if the given code in Program 10.6 is executed? Give a reason why such exception object is thrown.

[Objek pengecualian yang manakah yang akan dibuang apabila kod di dalam Aturcara 10.6 dilaksanakan? Berikan sebab kenapa objek dari kelas pengecualian tersebut dibuang.]

```

1 //Program 10.6
2 class excep
3 {
4     public static void main(String[]arg)
5     {
6         int a[]={5,10};
7         int b=5;
8         try
9         {
10            int x=a[2]/b-a[1];
11        }
12        catch(NumberFormatException e)
13        {
14            System.out.println(e);
15        }
16        catch(ArithmetricException e)
17        {
18            System.out.println("Division by Zero");
19        }
20        catch(Exception e)
21        {
22            System.out.println("Exception: "+e);
23        }
24        int y=a[1]/a[0];
25        System.out.println("y = "+y);
```

|    |   |
|----|---|
| 26 | } |
| 27 | } |

5. Compile and run Program 10.7. Understand what each line of the statement in the program does. In your own words, explain **why such output is printed**.  
*[Kompil dan laksanakan Aturcara 10.7. Fahamkan apa penyataan pada setiap baris di dalam aturcara tersebut lakukan. Di dalam ayat kamu sendiri, terangkan kenapa hasil tersebut dicetak.]*

```

1 //Program 10.7
2 public class ExceptionTest
3 {
4     public static void main(String[]a)
5     {
6         int number;
7         String str;
8
9         try
10        {
11            str="xyz";
12            number=Integer.parseInt(str);
13            System.out.println("A");
14        }
15
16        catch(NumberFormatException e)
17        {
18            System.out.println("B");
19        }
20
21        catch(IllegalArgumentException e)
22        {
23            System.out.println("C");
24        }
25        finally
26        {
27            System.out.println("D");
28        }
29
30        System.out.println("E");
31    }
32}

```

6. Compile and run Program 10.8. Understand what each line of the statement in the program does. In your own words, explain **clearly the throw Exception object concept as illustrated in the Program 10.8.**  
*[Kompil dan laksanakan Aturcara 10.8. Fahamkan apa penyataan pada setiap baris di dalam aturcara tersebut lakukan. Di dalam ayat kamu sendiri, terangkan dengan jelas konsep throw objek Exception yang diilustrasikan di dalam Aturcara 10.8.]*

|   |                     |
|---|---------------------|
| 1 | //Program 10.8      |
| 2 | import java.io.*;   |
| 3 | import java.util.*; |

```

4
5   class TestDivide
6   {
7     public static void main(String[]a) throws IOException,
8 ArithmeticException
9     {
10       Scanner in = new Scanner(System.in);
11
12       try
13       {
14         int denom = in.nextInt();
15
16         if (denom<=0)
17           throw new ArithmeticException("Input positive
18 denom");
19
20         System.out.println("Value      of      100/denom:
21 "+100/denom);
22       }
23       catch(ArithmeticException e)
24       {
25         System.out.println("Arithmetic      Exception
26 occurs!");
27       }
28
29       catch(NumberFormatException e)
30       {
31         System.out.println("Wrong data type");
32       }
33     }
34   }

```

1. Type in the Program 10.9.  
 [Taip Aturcara 10.9.]

```

1 //Program 10.9.
2 import java.io.*;
3 public class Exceptl
4 {
5   public static int num, denom;
6   public static double out=0.0;
7
8   public static void main(String[]a) throws IOException
9   {
10     BufferedReader      st=new      BufferedReader(new
11 InputStreamReader(System.in));
12     test1();
13   }
14
15   public static void test1()
16   {
17     try
18     {
19       test2();
20     }
21     catch(ArithmeticException ex)

```

```

22    {
23        System.out.println("Arithmetic problem!");
24        System.out.println(ex.getMessage());
25    }
26    catch(NumberFormatException ex)
27    {
28        System.out.println("Wrong data type");
29        ex.printStackTrace();
30    }
31    catch(IOException ex)
32    {
33        System.out.println("Input problem");
34    }
35}
36
37 public static void test2() throws IOException
38 {
39     BufferedReader st=new BufferedReader(new
40         InputStreamReader(System.in));
41     num=Integer.parseInt(st.readLine());
42     denom=Integer.parseInt(st.readLine());
43     out=num/denom;
44     System.out.println(out);
45 }
46 }
```

- i. Run the Program 10.10 with the input: 4 0. What is the result. In your own words, explain how such output is produced.  
*[Laksanakan Aturcara 10.10 dengan input: 4 0. Apakah hasilnya. Di dalam ayat kamu sendiri, terangkan bagaimana output tersebut terhasil.]*
- ii. Now, run the Program 10.10 with the input: r. What is the result? In your own words, explain how such output is produced.  
*[Laksanakan Aturcara 10.10 dengan input: r. Apakah hasilnya. Di dalam ayat kamu sendiri, terangkan bagaimana output tersebut terhasil.]*

8. Type in the Program 10.11

*[Taip Aturcara 10.11.]*

```

1 //Program 10.11
2 import java.io.*;
3
4 class Reader1
{
5
6     static BufferedReader st=new BufferedReader(new
7         InputStreamReader(System.in));
8     public static int readInt()
9     {
10        while(true)
11        {
12            try
13            {
14                return Integer.parseInt(st.readLine());
15            }
16        }
17    }
18}
```

```

4   class TestDivide
5   {
6       public static void main(String[]a) throws IOException,
7           ArithmeticException
8       {
9           Scanner in = new Scanner(System.in);
10      try
11      {
12          int denom = in.nextInt();
13
14          if (denom<=0)
15              throw new ArithmeticException("Input positive
16          denom");
17
18          System.out.println("Value      of      100/denom:
19          "+100/denom);
20
21      }
22      catch(ArithmeticException e)
23      {
24          System.out.println("Arithmetic      Exception
25          occurs!");
26      }
27
28      catch(NumberFormatException e)
29      {
30          System.out.println("Wrong data type");
31      }
32
33  }
34 }

```

7. Type in the Program 10.9.  
*[Taip Aturcara 10.9.]*

```

1 //Program 10.9.
2 import java.io.*;
3 public class Except1
4 {
5     public static int num, denom;
6     public static double out=0.0;
7
8     public static void main(String[]a) throws IOException
9     {
10        BufferedReader      st=new      BufferedReader(new
11        InputStreamReader(System.in));
12        test1();
13    }
14
15    public static void test1()
16    {
17        try
18        {
19            test2();
20        }
21        catch(ArithmeticException ex)

```

```

22     {
23         System.out.println("Arithmetic problem!");
24         System.out.println(ex.getMessage());
25     }
26     catch(NumberFormatException ex)
27     {
28         System.out.println("Wrong data type");
29         ex.printStackTrace();
30     }
31     catch(IOException ex)
32     {
33         System.out.println("Input problem");
34     }
35 }
36
37     public static void test2() throws IOException
38     {
39         BufferedReader st=new BufferedReader(new
40             InputStreamReader(System.in));
41         num=Integer.parseInt(st.readLine());
42         denom=Integer.parseInt(st.readLine());
43         out=num/denom;
44         System.out.println(out);
45     }
46 }
```

- i. Run the Program 10.10 with the input: **4 0**. What is the result? In your own words, explain how such output is produced.

[Laksanakan Aturcara 10.10 dengan input: **4 0**. Apakah hasilnya. Di dalam ayat kamu sendiri, terangkan bagaimana output tersebut terhasil.]

- ii. Now, run the Program 10.10 with the input: **r**. What is the result? In your own words, explain how such output is produced.

[Laksanakan Aturcara 10.10 dengan input: **r**. Apakah hasilnya. Di dalam ayat kamu sendiri, terangkan bagaimana output tersebut terhasil.]

8. Type in the Program 10.11

[Taip Aturcara 10.11.]

```

1 //Program 10.11
2 import java.io.*;
3
4 class Reader1
5 {
6     static BufferedReader st=new BufferedReader(new
7         InputStreamReader(System.in));
8     public static int readInt()
9     {
10        while(true)
11        {
12            try
13            {
14                return Integer.parseInt(st.readLine());
15            }
16        }
17    }
18 }
```

```

15     }
16     catch(Exception ex)
17     {
18         System.out.println("Invalid");
19     }
20 }
21
22
23     public static float readFloat() throws
24 NumberFormatException,
25 IOException
26 {
27     Float ob=new Float(st.readLine());
28     return(ob.floatValue());
29 }
30
31
32 public class Program1
33 {
34     public static void main(String[]a) throws IOException
35     {
36         try
37     {
38         System.out.println("Input 2 integers: ");
39         int int1=Reader1.readInt();
40         int int2=Reader1.readInt();
41         System.out.println("int1+int2= "+(int1+int2));
42         System.out.println("Input 2 floating numbers:
43 ");
44         float num1=Reader1.readFloat();
45         float num2=Reader1.readFloat();
46         System.out.println("num1+num2= "+(num1+num2));
47     }
48     catch(NumberFormatException e)
49     {
50         System.out.println("Exception occurs");
51     }
52     System.out.println("END");
53 }
54 }

```

- i. Run the Program 10.11 with the input in Figure 10.2. What is the result. In your own words, explain how such output is produced.  
*[Laksanakan Aturcara 10.11 dengan input di dalam Rajah 10.2. Apakah hasilnya. Di dalam ayat kamu sendiri, terangkan bagaimana output tersebut terhasil.]*

|       |
|-------|
| P     |
| 9     |
| q     |
| 10    |
| 40.0  |
| 50.75 |

Figure 10.2

- ii. Now, run the Program 10.11 with the input in Figure 10.3 . What is the result?  
In your own words, explain how such output is produced.

[Laksanakan Aturcara 10.11 dengan input di dalam Rajah 10.3. Apakah hasilnya. Di dalam ayat kamu sendiri, terangkan bagaimana output tersebut terhasil.]

|       |
|-------|
| 5     |
| 6     |
| w     |
| 40.65 |

9. Understand the following Program 10.13 and answer the questions below.  
[Fahamkan Aturcara 10.13 dan jawab soalan-soalan yang berikut.]

```

1 //Program 10.13
2 public class BadDivide
3 {
4
5     public void method1(int n)
6     {
7         method2(100, n);
8     }
9
10    public void method2(int n, int d)
11    {
12        System.out.println(n/d);
13    }
14
15    public static void main(String args[])
16    {
17        BadDivide bd = new BadDivide();
18        for (int k=0; k<5; k++)
19        bd.method1(k);
20    }
21 }
```

- i. Identify the line statement that will cause the divide-by-zero exception.  
[Kenal pasti penyataan di baris mana yang akan menyebabkan pengecualian divide- by-zero.]
- ii. Write the stack trace that will be printed by the above program.  
[Tulis paparan surihan yang akan dicetak oleh aturcara di atas.]
- iii. Modify method2() to perform divide-by-zero exception handling on its own.  
The modified method2() must be able to print error message and stack trace.  
[Ubah suai method2() untuk mengendalikan pengecualian divide-by-zero sendiri. Method2() yang telah diubah suai mesti mampu untuk mencetak mesaj ralat dan paparan surihan.]

0. Why does the following code cause a compiler error? Correct the code.

[Kenapa kod-kod berikut akan menyebabkan ralat pengkompil? Betulkan kod tersebut.]

i.

```
1 catch (FileNotFoundException e)
2 {
3     System.out.println("File Not Found");
4 }
5 try
6 {
7     File file=new File("myFile.txt");
8     Scanner inputFile=new Scanner(file);
9 }
```

ii.

```
1 //Assume inputFile references a Scanner object
2
3 try
4 {
5     input=inputFile.nextInt();
6 }
7 finally
8 {
9     inputFile.close();
10 }
11 catch (InputMismatchException e)
12 {
13     System.out.println(e.getMessage());
14 }
```

iii.

```
1 try
2 {
3     number=Integer.parseInt(str);
4 }
5 catch(Exception e)
6 {
7     System.out.println(e.getMessage());
8 }
9 catch (IllegalArgumentException e)
10 {
11     System.out.println("Bad Number Format");
12 }
13 catch (NumberFormatException e)
14 {
15     System.out.println(str + " is not a number");
16 }
```

iv.

```

1   try
2   {
3       number=Integer.parseInt(str);
4   }
5   catch(IllegalArgumentException e)
6   {
7       System.out.println(e.getMessage());
8   }
9   catch (IllegalArgumentException e)
10  {
11      System.out.println("Bad Number Format");
12  }
13  catch (NumberFormatException e)
14  {
15      System.out.println(str + " is not a number");
16  }

```

v.

```

1 class excep
2 {
3     public static void main(String[]arg)
4     {
5         int a[]={5,10};
6         int b=5;
7         try
8         {
9             int x=a[2]/b-a[1];
10        }
11        catch(NumberFormatException e)
12        {
13            System.out.println(e);
14        }
15        int y=a[1]/a[0];
16        System.out.println("y = "+y);
17        catch(ArithmetricException e)
18        {
19            System.out.println("Division by Zero");
20        }
21        catch(Exception e)
22        {
23            System.out.println("Exception: "+e);
24        }
25    }
26 }
27

```

## EXERCISE 4: THROW EXCEPTION

[LATIHAN 4: PENGECUALIAN THROW]

### CONCEPT [KONSEP]:

- 11.8 ‘**throw**’ keyword is used to throw the exceptions explicitly.  
 [Katakunci **throw** digunakan untuk membuang exception secara nyata.]

- Syntax for throw:

throw ThrowableInstance;

Example;

```

1 //Program 12.1
2 class ThrowDemo {
3     static void demoproc() {
4         try {
5             throw new NullPointerException("demo");
6         } catch (NullPointerException e) {
7             System.out.println("caught in demoproc");
8             throw e;
9         }
10    }
11    public static void main (String args[]) {
12        try {
13            demoproc();
14        } catch (NullPointerException e) {
15            System.out.println("recaught:" + e);
16        }
17    }
  
```

- 11.9 ‘**throws**’ keyword lists the type of exceptions that a method might throw.

[Katakunci **throws** menyenaraikan jenis exception yang berkemungkinan dibuang oleh sesuatu metod.]

- Syntax for throws:

```
type method-name(parameter-list) throws exception-list
{
    //body of method
}
```

Example;

```

1 //Program 12.1
2 class ThrowsDemo {
3     static void throwOne() throws IllegalAccessException {
4         System.out.println("Inside throwOne.");
5         throw new IllegalAccessException("demo");
6     }
7     public static void main(String args[]) {
8         try {
9             throwOne();
10        } catch (IllegalAccessException e) {
11            System.out.println("Caught" + e);
12        }
13    }
  
```

1. What is necessity to re-throw an exception  
[Kenapa perlu re-throw exception?]
2. Compile and run Program 10.3 with the input as in Figure 10.2. Understand what each line of the statement in the program does. In your own words, explain **why such output is printed**. Write the **expected stack trace** that will be printed as an output. **Explain the meaning** of the printed stack trace.  
[Kompil dan laksanakan Aturcara 10.3 dengan memasukkan data seperti di dalam Rajah 10.1. Fahamkan apa penyataan pada setiap baris di dalam aturcara tersebut lakukan. Di dalam ayat kamu sendiri, terangkan **kenapa hasil tersebut dicetak**. Tuliskan **hasil surihan** yang dijangka akan dicetak sebagai output. Terangkan **maksud hasil surihan yang dicetak**.]

```

1 //Program 10.3
2 import java.util.*;
3 import java.io.*;
4
5 public class TestSum
6 {
7     public static void main (String[]a) throws IOException
8     {
9         double num[]=new double[5];
10        double total=0;
11
12        Scanner in = new Scanner(System.in);
13
14        System.out.print("How many floating point numbers do
15 you want
16            to input?");
17        int max= in.nextInt();
18
19        System.out.println("Enter "+max+" numbers");
20
21        for(int i=0;i<max;i++)
22        {
23            num[i]=in.nextDouble();
24            total+=num[i];
25        }
26        System.out.println("End of Program");
27        System.out.println("Total: "+total);
28    }
29 }
```

|       |
|-------|
| 6     |
| 4.25  |
| 5.55  |
| 6.35  |
| 7.85  |
| 8.90  |
| 9.33  |
| 10.80 |

Figure 10.2

**EXERCISE 5: CREATE OWN EXCEPTION**  
**[LATIHAN 5 : MEMBINA PENGEJUALIAN SENDIRI]**

**CONCEPT [KONSEP]:**

- 11.10 You can create your own exceptions, normally by extending Exception or one of its subtypes.  
*[Kamu boleh mencipta kelas pengejualian sendiri dengan 'extend' daripada kelas Exception.]*

1. Type in the Program 10.11 and Program 10.12  
*[Taip Aturcara 10.11 dan Aturcara 10.12]*

```

1 //Program 10.11
2 public class MyException extends Exception
3 {
4     private int value;
5     private String msg;
6
7     public MyException(int i)
8     {
9         value=i;
10        msg=new String("Exception happens at value:
11 "+value);
12    }
13
14     public String getMessage()
15     {
16         return msg;
17     }
18
19     public void printStackTrace()
20     {
21         System.out.println("The problem is with the value:
22 "+value);
23     }
24 }
```

```

1 //Program 10.12
2 public class Propagate
3 {
4     public void method1(int i)
5     {
6         try
7         {
8             method2(i);
9         }
10
11         catch(MyException e)
12         {
13             e.printStackTrace();
14         }
15     }
16 }
```

```

17     public void method2(int i) throws MyException
18     {
19         if(i==0)
20             throw new MyException(i);
21         else
22             System.out.println("i: "+i);
23     }
24     public static void main(String[]a)
25     {
26         Propagate p=new Propagate();
27         for(int i=0;i<3;i++)
28             p.method1(i);
29     }
30 }
31 }
```

Run the above program. What is the output. Explain in details the steps involve until the output is printed.

[Laksanakan aturcara di atas. Apakah hasilnya? Terangkan dengan terperinci langkah-langkah yang terlibat sehingga output tersebut terhasil.]

2. Custom exception has been created in the Program 10.14 below. Correct the code.  
[Pengecualian khas telah dicipta di dalam Aturcara 10.14. Betulkan kod tersebut.]

```

1 //Program 10.14
2 import java.util.*;
3 class myexception extends Exception
4 {
5     myexception(String s)
6     {
7         super(s);
8     }
9 }
10 class excep4
11 {
12     public static void main(String a[])
13     {
14         String st1;
15
16         Scanner in = new Scanner(System.in);
17         st1=in.nextLine();
18         if (st1=="Hello")
19             System.out.println("String is right");
20         else
21             try
22             {
23                 throw new myexception("Invalid String");
24             }
25             catch(myexception ex)
26             {
27                 System.out.println(ex.getMessage());
28             }
29     }
30 }
31 }
```

**EXERCISE 6: PROBLEM SOLVING**  
[LATIHAN 6 : PENYELESAIAN MASALAH]

1. Write a class with a static method that accepts characters one at a time, counts the number of characters and throws an exception if a character is not in the set{'a'..'z','0'..'9','A'..'Z'}. The exception should be thrown but not caught (i.e., no explicit catch block). Write a test program that calls this method and prints "Error in Input" if the method throws an exception.  
[Tulis satu kelas yang mempunyai metod static yang menerima aksara satu persatu, bilang aksara tersebut dan throws exception jika aksara bukan dalam set {‘a’..‘z’,‘0’..‘9’,‘A’..‘Z’}. Exception hendaklah dibuang tidak dikendalikan (iaitu tanpa blok catch). Tulis aturcara Ujian yang memanggil metod ini dan mencetak “Ralat Pada Input” jika metod ini membuang satu exception.]
2. Write a Java program that will declare an array of five integers and store five values in the array. Write a try block in which it will display the contents of the array. Create a catch block that catches the `ArrayIndexOutOfBoundsException` object if the program tries to display the sixth element and display the message, “You have exceeded the array index”.  
[Tulis satu aturcara Java yang mengisyiharkan satu tatasusunan 5 integer dan akan menyimpan 5 nilai. Tulis satu blok try yang akan memaparkan kesemua kandungan tatasusunan tersebut. Cipta satu blok catch yang akan menangkap objek `ArrayIndexOutOfBoundsException` sekiranya aturcara tersebut cuba untuk memaparkan nilai yang keenam dan paparan “Anda telah melebihi indeks tatasusunan” akan dicetak.]
3. Write a Java program that tries to convert a String to an integer (hint: you can use the `Integer.parseInt()` method). A `NumberFormatException` object will be thrown if the user tries to convert a String that cannot be represented as an integer value (for example, “A”). Your program will also display an appropriate error message.  
[Tulis satu aturcara Java yang cuba menukarkan satu rentetan kepada integer (tip: Kamu boleh menggunakan metod `Integer.parseInt()`). Satu objek `NumberFormatException` akan dibuang sekiranya pengguna cuba untuk menukarkan satu rentetan yang tidak boleh diwakilkan sebagai nilai integer (seperti “A”). Aturcara kamu juga akan memaparkan mesej ralat yang bersesuaian.]
4. Write a Java program that calculates a square root of a value. Prompt the user for an input value and use the `Math.sqrt()` method on it. An `ArithmaticException` object will be thrown if the user tries to calculate a square root of a negative value. Your program will either display the square root or catches the thrown `ArithmaticException` object and displays an appropriate message.  
[Tulis satu aturcara Java yang mengira punca kuasa dua satu nilai. Arahkan pengguna untuk memasukkan satu nilai dan gunakan metod `Math.sqrt()` ke atas nilai tersebut. Satu objek `ArithmaticException` akan dibuang sekiranya

pengguna cuba untuk mengira punca kuasa dua satu nilai negatif. Aturcara kamu akan samada memaparkan nilai punca kuasa dua atau menangkap objek ArithmeticException yang dibuang dan memaparkan mesej ralat yang bersesuaian.]

5. Rewrite the Final class in Program 10.13 in the try{}, catch{} and finally{} hierarchy form to manage the exceptions if the file does not exist.  
[Tulis semula kelas Final di Aturcara 10.13 dalam bentuk blok try{} catch{} finally{} berhiraki untuk menguruskan ralat yang timbul akibat fail tidak ditemui.]

```
1 //Program 10.14
2 import java.io.*;
3 class Final
4 {
5     public void readFile() throws IOException
6     {
7         FileReader reader = new FileReader
8 ("c:\\windows\\data.dat");
9         int temp = reader.read();
10        System.out.println(temp);
11        reader.close();
12        System.out.println("operasi tamat");
13    }
14 }
15 }
```

6. As a software consultant, Dr. Mumtaz is requested by one insurance company to develop a software that can calculate the age of a client by subtracting year birth (read from keyboard) from current year (2009). The program can compute the number of years the person has been driving by subtracting 16 from his/her age. Client that drive less than 4 years will be charged RM1000 per year. Otherwise the charge is RM600. However if his/her age is less than 16 years, an exception object is thrown. All the input & output statement must be placed in one block try and the program must catch exception.  
[Sebagai seorang penasihat perisian, Dr. Mumtaz diminta oleh sebuah syarikat insuran untuk membangunkan sebuah perisian yang dapat mengira usia seseorang pelanggan dengan menolak tahun lahir (dibaca melalui papan kekunci) daripada tahun semasa (2008). Aturcara tersebut akan mengira bilangan tahun pelanggan tersebut telah memandu dengan menolak 16 daripada umur nya. Pelanggan yang memandu kurang daripada 4 tahun akan di caj RM1000 setiap tahun. Kalau tidak, caj nya ialah RM600. Tetapi, sekiranya umur nya kurang daripada 16 tahun, satu objek pengecualian akan dibuang. Semua penyataan masukan dan keluaran mesti diletak di dalam satu blok try dan aturcara tersebut akan mengendalikan objek pengecualian yang dibuang.]
7. Write an application that displays a series of at least 10 student ID numbers (use an array to store these data) and asks the user to enter a final grade for the OOP subject

for each of the student. Create an Exception class named GradeException that contains a static public array of valid grade letters ('A+', 'A', 'A-', 'B+', 'B', 'B-', 'C', 'D', 'F', 'I'), which you can use to determine whether a grade entered from the application is valid. In your application, throw a GradeException object if the user does not enter a valid letter grade. Catch the GradeException object and then display an appropriate message. In addition, store an 'I' (for Incomplete) for any student for whom an exception is caught. At the end of the application, display all the students IDs and OOP final grades.

[Tulis satu aturcara yang akan memaparkan sekurang-kurangnya 10 nombor ID pelajar (guna kan tatasusunan untuk menyimpan data-data ini) dan arahkan pengguna untuk memasukkan gred akhir untuk matapelajaran OOP untuk setiap pelajar ini. Cipta satu kelas Exception bernama GradeException yang mengandungi satu tatasusunan abjad gred yang sah ('A+', 'A', 'A-', 'B+', 'B', 'B-', 'C', 'D', 'F', 'I'). Tatasusunan ini mestilah statik dan bercapaian public, dan boleh dikenalpasti oleh aturcara samada gred yang dimasukkan itu sah atau tidak. Di dalam aturcara kamu, objek GradeException akan dibuang sekiranya pengguna tidak memasukkan gred yang sah. Objek GradeException ini akan dikendalikan dan akan memaparkan mesej yang bersesuaian. Sebagai tambahan, 'I' (untuk Tidak Tamat) akan disimpan untuk mana-mana pelajar yang objek pengecualiannya ditangkap. Di akhir aturcara, paparkan kesemua ID pelajar dan gred akhir mereka.]

- Answer questions below based on the class diagram in Figure 10.5.

[Jawab soalan-soalan yang berikut berdasarkan rajah kelas di dalam Rajah 10.5.]

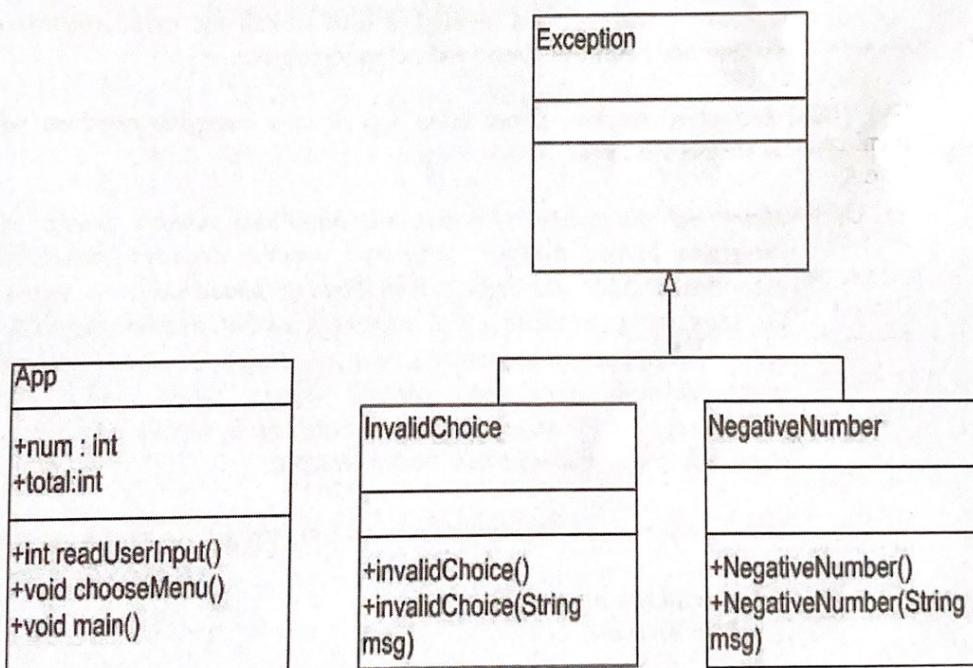


Figure 10.5: Class Diagram

- i. Write a Java program for `InvalidChoice` and `NegativeNumber` class. Both classes are subclasses to `Exception` class.

[Bina kod aturcara Java untuk kelas `InvalidChoice` dan `NegativeNumber` di atas yang merupakan subkelas kepada `Exception`.]

- ii. Write a Java program for `App` class based on the information below:

- a. `readUserInput()` method: This method is used to read an integer number from a user until successful. If a negative number is entered, the `NegativeNumber` exception object will be thrown. The `readUserInput()` method will handle the `NegativeNumber` exception object. Besides that, this method will also handle an exception object resulted from invalid input. The method will calculate a total of integers entered and display the total.
- b. `chooseMenu()` method: this method is used to choose an operation:
1. enter an integer.
  2. quit from the program.

If the user enters other than above, the method will throw a `InvalidChoice` exception object. The method will not handle the `InvalidChoice` exception object BUT the method will throw the `InvalidChoice` exception object back to the caller.

- c. `main()` method: this method is used to call the `chooseMenu()` method and handle any received exception objects.

[Bina kod aturcara Java untuk kelas `App` di atas mengikut panduan yang diberikan seperti berikut:

- a. Metod `readUserInput()` untuk membaca satu nombor integer dari pengguna hingga berjaya. Sekiranya nombor negative dimasukkan, exception `NegativeNumber` akan dibuang. Metod `readUserInput()` ini akan mengendalikan objek exception akibat nombor negatif ini. Selain daripada itu, metod `readUserInput()` ini juga akan mengendalikan exception akibat input tidak sah. Metod `readUserInput()` seterusnya akan menjumlah semua nombor yang dimasukkan dan memaparkan jumlah tersebut.

- b. Metod `chooseMenu()` untuk membenarkan pengguna memilih operasi:

- 1: memasukkan input.
- 2: keluar aturcara

*Jika input selain di atas, chooseMenu() akan membuang objek pengecualian InvalidChoice. Metod chooseMenu() tidak akan mengendalikan InvalidChoice, sebaliknya metod tersebut akan membuang objek pengecualian InvalidChoice semula kepada pemanggil.*

- c. *Metod main() untuk memanggil metod chooseMenu() dan mengendalikan objek pengecualian yang diterimanya.]*
9. The program below is possibly raises several types of errors depending on the input given by the user. Read carefully and answer all the questions below:  
*[Aturcara di bawah berkemungkinan mengeluarkan beberapa ralat bergantung kepada nilai masukan pengguna. Baca dan jawab semua soalan di bawah:]*
- i. Name all the errors that possibly occur.  
*[Namakan ralat-ralat yang berkemungkinan berlaku.]*
  - ii. Rewrite the code and add related exception handlers to catch the errors.  
*[Tulis semula aturcara beserta dengan pengendali-pengendali ralat bagi memerangkap ralat- ralat.]*
  - iii. Write your own exception handler named “newException” to raise an error when the input number is 5.  
*[Tulis satu pengendali ralat anda bernama “NewException” bagi mengeluarkan ralat jika pengguna memasukkan nilai 5.]*

```

1 //Program 10.15
2 import java.util.Scanner;
3
4 public class testException
5 {
6     public static void main (String args[ ])
7     {
8         int array[] = {2,3,4,5,6,7,8,9};
9         int result = 0;
10        Scanner scanner = new Scanner(System.in);
11        System.out.print("Enter an integer: ");
12        int number = scanner.nextInt();
13
14        result = 100/number;
15        System.out.println (100/array [number]);
16        System.out.println ("The result is : "
17 "+result);
18    }
19 }
```