Tutorial 4 : Exception Handling

1. Output for waitTime = 46:
   Try block entered
   Exception: Time Limit Exceeded
   After catch block

   Output for waitTime = 12:
   Try block entered
   Leaving try block
   After catch block

2. The code is given below:

```
public class PowerFailureException extends Exception {
    public PowerFailureException()
    {
        super("Power Failure!");
    }
    public PowerFailureException(String message)
    {
        super(message);
    }
}
```

3. Output for argument 99:
   In finally block.
   Caught in main.

   Output for argument -99:
   Caught in sampleMethod.
   In finally block.
   After finally block.

   Output for argument 0:
   No Exception.
   Still in sampleMethod.
   In finally block.
   After finally block.

4. The program is given below:

```
public class UnknownOperatorException extends Exception
{
    public UnknownOperatorException( )
    {
        super("UnknownOperatorException");
    }
    public UnknownOperatorException(char op)
    {
        super(op + " is an unknown operator.");
    }
    public UnknownOperatorException(String message)
    {
        super(message);
    }
}
```

```
/* Calculator Application */
import java.util.Scanner;
public class CalculatorEx
{
    private double result;
    private double precision = 0.0001;
    // Numbers close to zero are treated as if equal to zero.

    public static void main(String[] args)
    {
        CalculatorEx cal = new CalculatorEx();
        try
      {
            System.out.println("Calculator is on");
            cal.doCalculation();
        }
        catch(UnknownOperatorException e)
        {
            cal.handleUnknownOpException(e);
        }
        catch(Exception e)
        {
            System.out.println("Other Exception " + e.getMessage());
         }

        System.out.println("Final result = " + cal.resultValue());
        System.out.println("End of program");
    }
    public CalculatorEx()
    {
        result = 0;
    }
    public double resultValue()
    {
        return result;
    }
     public void doCalculation() throws ArithmeticException,
       UnknownOperatorException
    {
        char nextOp;
        double nextNumber;
        boolean done = false;
        Scanner sc = new Scanner(System.in);

        System.out.println("result = " + result);
        while (!done)
        {
            System.out.println("Enter + - * / or Q/q to quit");
            String nextOpStr = sc.next();
            nextOp = nextOpStr.charAt(0);
            if ((nextOp == 'Q') || (nextOp == 'q'))
                done = true;
            else
            {
               System.out.println("Enter the number > ");
               nextNumber = sc.nextDouble();
               result = evaluate(nextOp, result, nextNumber);
               System.out.println("result " + nextOp + " "
                               + nextNumber + " = " + result);
               System.out.println("updated result = " + result);
            }
        }
    }
```

```
    public double evaluate(char op, double n1, double n2)
              throws ArithmeticException, UnknownOperatorException
    {
        double answer;
        switch (op)
        {
            case '+':
                answer = n1 + n2;
                break;
            case '-':
                answer = n1 - n2;
                break;
            case '*':
                answer = n1 * n2;
                break;
            case '/':
                if ( (-precision < n2) && (n2 < precision))
                    throw new ArithmeticException();
                  answer = n1/n2;
                break;
            default:
                throw new UnknownOperatorException(op);
        }
        return answer;
    }

    public void handleUnknownOpException(UnknownOperatorException e)
    {
        System.out.println(e.getMessage( ));
        System.out.println("Please reenter:");
        try
        {
            doCalculation();
        }
        catch(UnknownOperatorException e2)
        {
            System.out.println(e2.getMessage( ));
            System.out.println("Try again later");
            System.out.println("End of Program");
            System.exit(0);
         } /*catch(Exception e1) {
            System.out.println("Other Exception " + e1.getMessage());
         }*/
    }

}
```