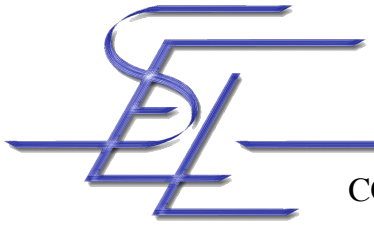
# Course Overview

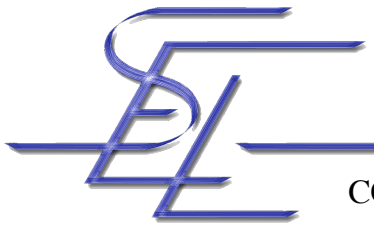
Jörg Kienzle
Software Engineering Laboratory
School of Computer Science
McGill University



McGill

# Outline

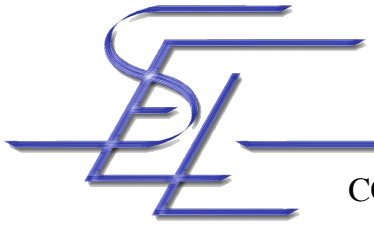
- Motivation
- Course Goals
- Course Information
- Background on me
- Suggested Textbooks
- Grading
- Questionnaire



McGill

# Motivation

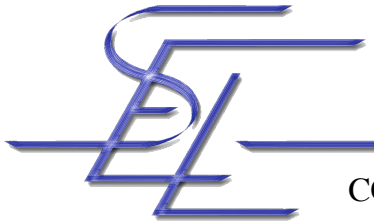
- Scope, complexity and pervasiveness of computer-based and controlled systems continue to increase
- Software assumes more and more responsibility
- Consequences of systems failing
  - Annoying to catastrophic
  - Opportunities lost, businesses failed, security breaches, systems destroyed, lives lost



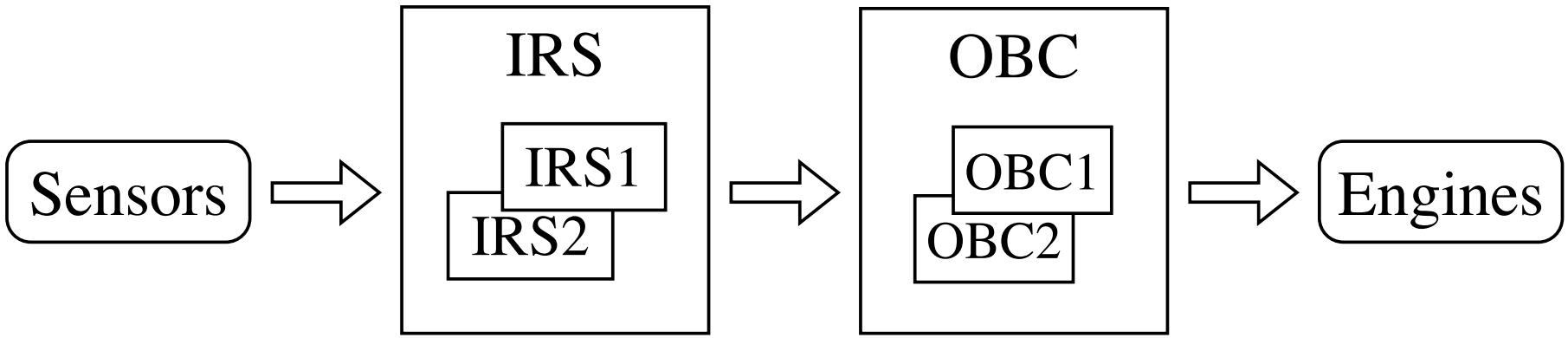
McGill

# Ariane V Disaster

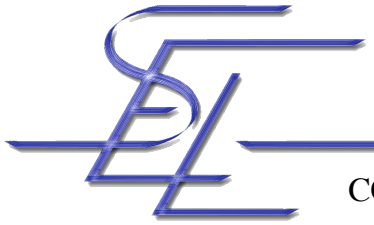
On June 4, 1996 an Ariane V rocket launched by the European Space Agency exploded just forty seconds after lift-off



# Ariane V Architecture




"hot standby"



McGill

# Ariane V Launch, June 4th 1996

IRS raises an *Operand Error* exception while converting a 64bit float to 16bit integer

No specific exception handler

*Operand Error* caused by high value of *Horizontal Bias*, which is normal for Ariane V

Function serves no purpose after lift-off in Ariane 5

Ariane IV, from which the code was reused, needs it during 50 seconds

Not possible to switch to backup IRS, for it had failed as well (72ms earlier)

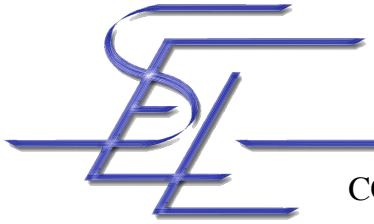
On-board Computer interprets “core dump” data as normal flight data

Full nozzle deflection of solid boosters and vulcan engine

Angle of attack > 20°

Separation of boosters from main stage

Self-destruction after 39 seconds



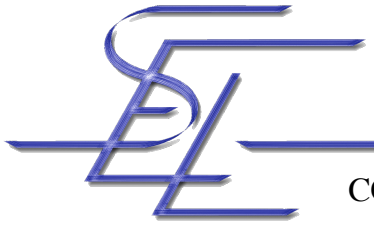
McGill

# Course Goals

- Goal of Fault Tolerance:

  Continue service in spite of design faults and faults of the surrounding environment

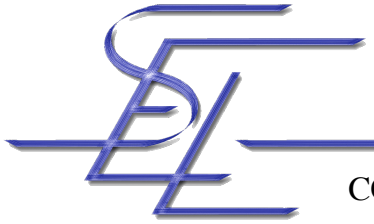
- Understand the important and fundamental concepts of fault tolerance
  - Be aware of the main techniques that can be applied by developers to produce fault-tolerant software
- Gain experience in implementation
  - Master the tricky subtleties of your favorite programming language



McGill

# Tentative Course Outline

- Fundamental Fault Tolerance Concepts
  - Terminology, Definitions, Fault Tolerance Context
- Programming language features for implementation
- Sequential Design Diverse Systems
  - Recovery blocks, retry blocks
- Independent Concurrent Systems
  - N-Version programming, N-Copy programming
- Dependability-Focused Requirements Elicitation
- Other Concurrent Fault Tolerance Techniques
  - Competitive: Transactions, etc…
  - Cooperative: Atomic Actions, etc…
  - Hybrid models



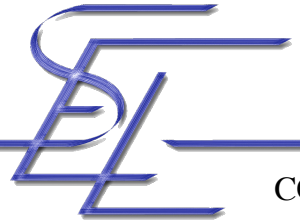
McGill

# Course Info

- Pre-requisites:
  COMP-409 Concurrent Programming
- Course hours:
  - Monday, Wednesday: 10:00 - 11:30
- Course webpage:

  http://www.cs.mcgill.ca/~joerg/SEL/COMP-667_Home.html

(handouts available for download there)



McGill

# Instructor

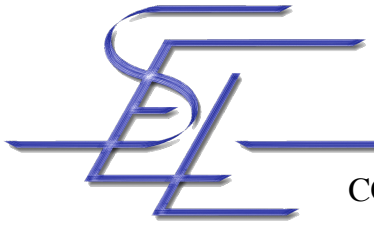
Jörg Kienzle
McConnell Engineering, room 327
Email: Joerg.Kienzle@mcgill.ca
Phone: (514) 398-2049
Home: (514) 871-2780

Office hours:

Monday: 11:30 - 12:30

+ any other time

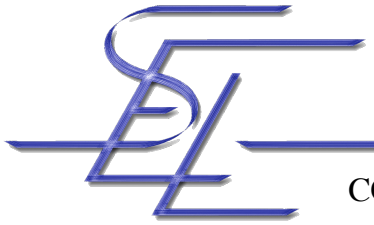
(send email)



McGill

# My Background

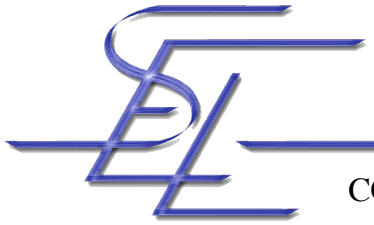
- Born in Princeton, NJ, USA
- German parents
- Grown up in Switzerland (German speaking part)
- Studied at the Swiss Federal Institute of Technology, Lausanne (French speaking part)
- Married a Canadian girl



McGill

# My Interests

- Fault tolerance
  - Integrating the concern of fault tolerance into the software development cycle
  - Determine the need for fault tolerance at the analysis level
  - Choose an appropriate architecture and fault tolerance model during design
  - Providing fault tolerance to the programmer (frameworks, aspect-orientation)
  - Implementing fault tolerance models on top of COTS middleware
- Fault tolerance in massively multi-player games
- Aspect-oriented Software Development

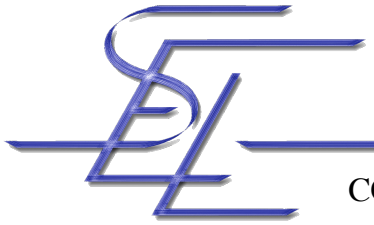


McGill

# Teaching Assistant

Wisam Al Abed
McConnell Engineering, room 322
Email: wisam.alabed@mail.mcgill.ca
Phone: (514) 398-7071 ext. 00116

Office hours:
TBA
(or by appointment)

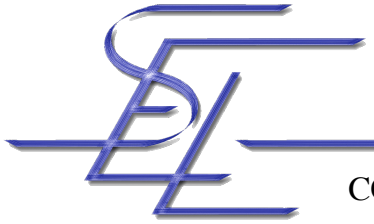


McGill

# Textbooks that Could Help

- Laura L. Pullum:
  *Software Fault Tolerance: Techniques and Implementation*, Artech House, Boston, 2001.
  ISBN: 1-58053-137-7

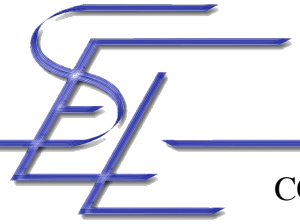
  - Available online at: http://library.books24x7.com/book/id_3628/toc.asp
- Jörg Kienzle:
  *Open Multithreaded Transactions: A Transaction Model for Concurrent Object-Oriented Programming*, Kluwer Academic Publishers, 2003.
  ISBN: 1-4020-1727-8



McGill

# More Books


- Jean-Claude Geffroy and Gilles Motet: *Design of Dependable Computing Systems*, Kluwer Academic Publishers, 2002. ISBN: 1-4020-0437-0
- P. A. Lee and T. Anderson: *Fault Tolerance - Principles and Practice*, 2nd edition, Springer Verlag, 1990. ISBN: 0-3878-2077-9
- K. Ramamritham and P. K. Chrysanthis: *Advances in Concurrency Control and Transaction Processing*, ACM Press, Los Alamitos, California, 1997. ISBN: 0-8186-7405-9

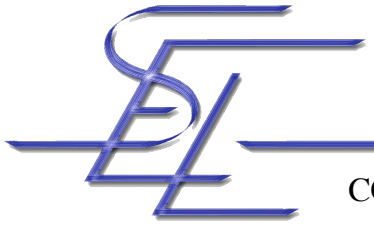


McGill

# Grading

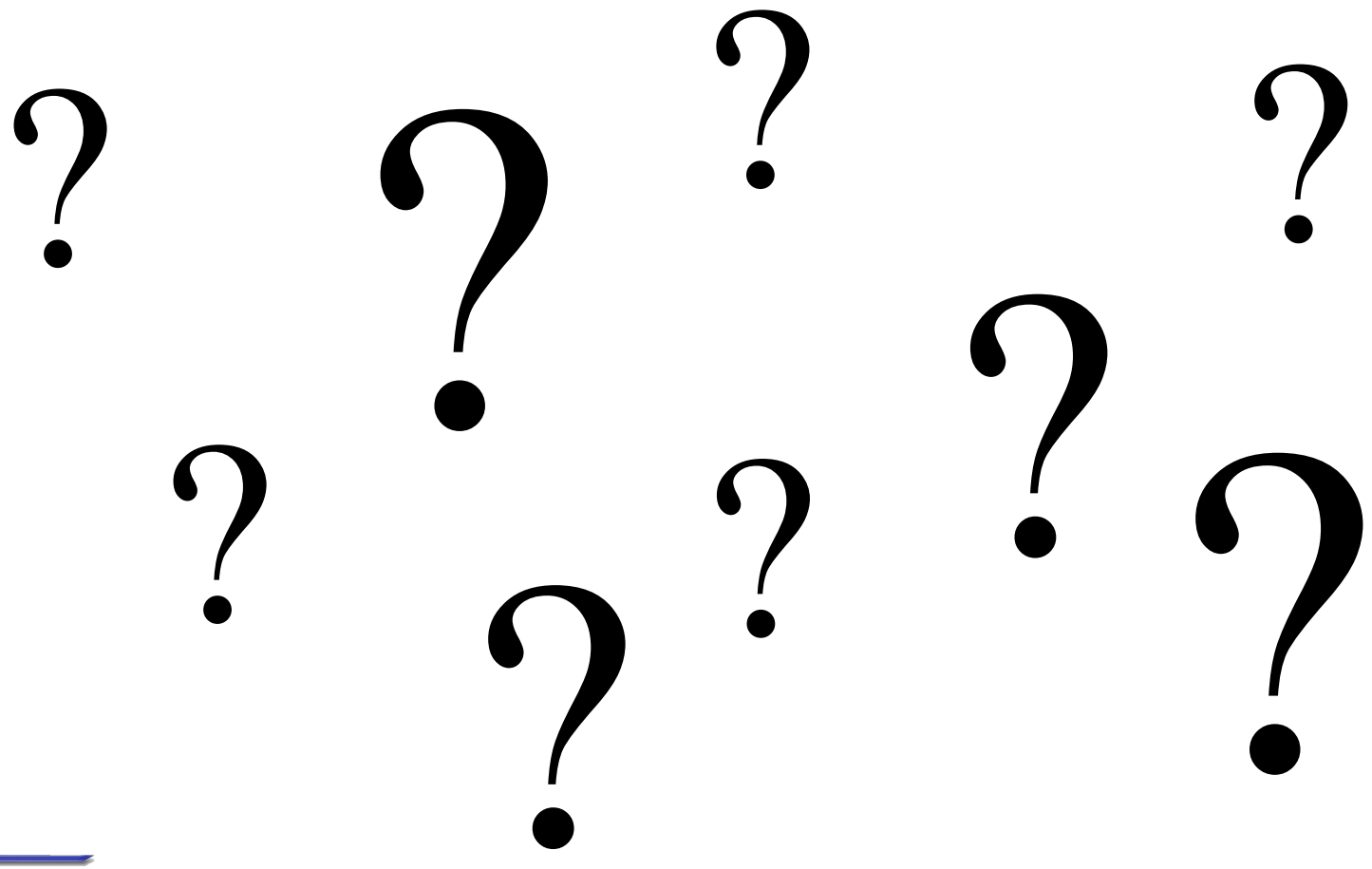
- 4 homework assignments
  - 1 warm-up assignment (5%)
  - 2 programming assignments (2 x 20% or (1x25% and 1x15%))
  - 1 non-programming assignment (1 x 10%)

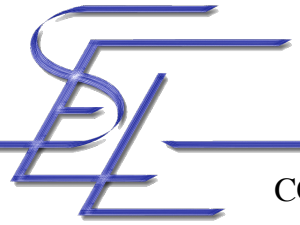
- Project (45%) (individual)
  - Provide a software fault tolerance scheme (distributed or concurrent) as a library / framework for a programming language of your choice
    Hand-in: short report and code
  - Study a specific software fault tolerance scheme or application using software fault tolerance (e.g. airbus, space-shuttle, TGV, air-traffic control, nuclear power plant, etc…)
    Hand-in: 45 - 60 min. presentation in class
  - Custom :)



McGill

# Questions?

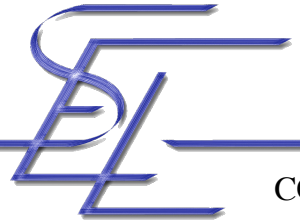




McGill

# Questionnaire

- For you
  - Evaluate your “concurrent” knowledge
- For me
  - To help me plan the course
  - Programming language background
  - Programming project or case study presentation?
- For all
  - Have some fun!



McGill