



"Business Programming"

**Critical Factors from Zero to Portable GUI Programming in
Four Hours**

BEE 2021 Conference

6th Business & Entrepreneurial Economics

Plitvice Lakes National Park, Croatia

May 26th – May 29th 2021

Rony G. Flatscher, WU (rony.flatscher@wu.ac.at)

Günter Müller, University of Freiburg (mueller@iig.uni-freiburg.de)



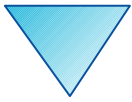
Challenges

- Teaching BA students with and without skills
- One semester, four hours, eight ECTS (200 hrs)
- Object-oriented programming
- Middle of semester: Windows and MS Office
- End of semester: Windows, Linux, MacOS and portable (!) GUIs, Internet programming, OpenOffice/LibreOffice, parsing XML text



Critical Success Factor

- CSF # 1: *Use an easy to learn programming language!*
 - ooRexx (acronym for "open object Rexx")
 - Human-centric design
 - Easy syntax, reads almost like pseudo code
 - Incorporates object-oriented concepts to "play with"
 - Developed originally by IBM handed over to RexxLA
 - Professional and powerful programming language
 - Open-source and free
 - Available for all major platforms (Windows, Linux, Apple)



Critical Success Factor

Example

```
-- a loop
do i=1 to 99          -- loop 99 times
  if i=1 then iterate -- next loop, skip remaining body
  if i=4 then leave  -- leave loop prematurely
  say "hello #" i    -- says "hello # 2" and "hello # 3"
end

-- demo arithmetics
say "i="i "1/i="1/i  -- says "i=4 1/i=0.25"

-- demo message
say "hi, there!"~reverse -- says "!ereht ,ih"

-- demo use of a class
say ".demo:" .demo    -- says ".demo: The DEMO class"
o=.demo~new           -- create an instance
say "o:" o             -- says "o: a DEMO"
o~name="BEE 2021"     -- assign value to attribute
say "o~name:" o~name  -- says "o~name: BEE 2021"
say "o~reverse:" o~reverse -- says "o~reverse: 1202 EEB"

-- demo defining a class
::class demo          -- defines the class DEMO
::attribute name      -- defines an attribute NAME
::method reverse      -- defines a method REVERSE
  return self~name~reverse -- returns the name reversed
```

```
hello # 2
hello # 3
```

```
i=4 1/i=0.25
```

```
!ereht ,ih
```

```
.demo: The DEMO class
```

```
o~name: BEE 2021
o~reverse: 1202 EEB
```

```
hello # 2
hello # 3
i=4 1/i=0.25
!ereht ,ih
.demo: The DEMO class
o: a DEMO
o~name: BEE 2021
o~reverse: 1202 EEB
```



Critical Success Factor

- CSF # 2: *Pareto principle*
 - Impossible to teach everything in detail
 - Teach conceptual, overview knowledge
 - Select the most "important concepts"
 - E.g. object-oriented paradigm, COM/OLE on Windows and MS Office, Java interface to become able to become portable (Windows, Linux, MacOS), GUI, Internet Programming, OpenOffice/LibreOffice, parsing XML text
 - Teach 80% of the most important concepts in 20% of the time
 - Rather than targeting 100% which would impose additional 80% of time, which is not available



Critical Success Factor

- CSF # 3: *Humboldt's ideal*
 - Observe the students
 - What problems do they get and why?
 - If necessary
 - Create new paths to ease understanding
 - Retest them in the next installment
 - Will allow to improve the course over time



Critical Success Factor

- CSF # 4: *No student is left alone*
 - Create groups of two students
 - Mix skills
 - A skilled student becomes "buddy tutor" for a
 - Zero-skilled student in that group



Critical Success Factor

- CSF # 5: *Searching the Internet*
 - Modern programming is about searching the Internet!
 - Find one own's coding problems and possible solutions
 - Find additional learning resources in all media forms on the Internet
 - E.g. tutorials for concepts, that are not yet understood, Youtube-videos for demonstrating the handling of development tools



Critical Success Factor

- CSF # 6: *Nutshell examples*
 - Make it as easy as possible to learn programming
 - Use easy to understand, small ("nutshell") programs
 - Show the output of nutshell programs on the slides
 - "Seeing is better than believing"



Critical Success Factor

- CSF # 7: *Weekly coding assignments*
 - Create two short (!) programs together in the group
 - Can help each other
 - Newbies can handle that and normally also are able to understand short programs from other groups on their own



Critical Success Factor

- CSF # 8: *Concluding project assignment*
 - Students suggest three projects combining
 - Three Windows programs ("Business Programming 1")
 - Three Java class libraries ("Business Programming 2")
 - One project will be picked and realized within a week
 - Project gets presented and demonstrated
 - Students experience success and their acquired skills



Business Programming 1 (BP1)

First Half of the Semester

- Statement, comment, symbol, variable, block, comparison, branch, loop
- Routine, label, scope, function, associative arrays
- Exception, handler, routine and requires directive, arguments by reference
- OO: Abstract datatype (ADT), class, attribute, method, creating an object/instance/value, message
- Class hierarchy, inheritance, collection classes and iteration
- Windows: COM, OLE, Windows registry, ooRexx class OLEObject to camouflage Windows, MS Excel, MS Word as ooRexx
- Windows: MS InternetExplorer (DHTML, fundamentals of HTML)



Business Programming 2 (BP2)

Second Half of the Semester

- Introduction to Java and BSF4ooRexx (camouflages Java as ooRexx)
- GUI concepts with events (and callbacks), Socket programming (client/server)
- OpenOffice/LibreOffice: UNO architecture, swriter, scalc, simpres
- XML: concepts, using SAX and DOM to parse XML text files efficiently
- Java scripting framework: BSF4ooRexx' REXXScriptEngine (allows ooRexx to be used as a Java scripting language in all Java applications)
- JavaFX: concepts, creating most complex GUIs in an easy manner
- Java module system, if students interested then .Net/CLR (support included in BSF4ooRexx)

Roundup

- Critical success factors
 - Regarded to be the most important ones
 - More success factors (e.g. e-Mail list, FB groups for help)
- Most important concepts
 - Business programming 1 (BP1)
 - Introduction to OO programming and COM, OLE (Windows)
 - Business programming 2 (BP2)
 - Introduction to Java and BSF4ooRexx, creating portable software that runs on Windows, Linux, MacOS