Platform-Based Development: Course Goals, Organization, Policies and Sample Topics

BS UNI studies, Spring 2024/2025

Dr Veljko Pejović
Veljko.Pejovic@fri.uni-lj.si

UNIVERSITY OF LJUBLJANA
FRI Faculty of Computer and Information Science

Instructor

- Dr Veljko Pejović
  - PhD in resource-efficient wireless networks (UCSB, USA)
  - Postdoc working on mobile sensing, human behaviour inference (Uni. Birmingham, UK)
  - Current projects and research interests:
    - Resource-efficient approximate mobile computing
    - Modelling user behaviour using mobile sensors (mostly security aspects)
  - Veljko.Pejovic@fri.uni-lj.si (“PBD 63287” in the subj.)
    - Use Slack for questions of general interest

Teaching Assistants

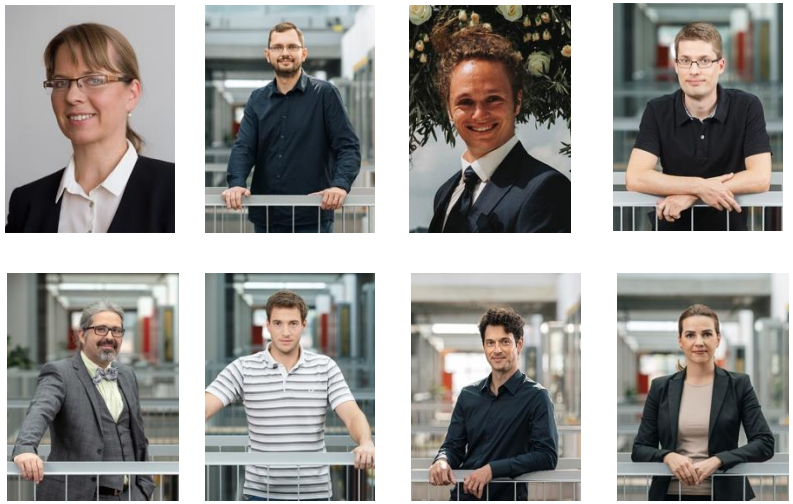
- Dr Octavian Machidon
  - PhD on reconfigurable computing from Transilvania University of Brasov, Romania
  - Research on ubiquitous computing, embedded systems, UAVs for agriculture
  - Octavian.Machidon@fri.uni-lj.si

Teaching Assistants

- Janez Božič
  - MS on federated learning on Android from UL FRI
  - Practical experience with Android programming, mobile/embedded system building, energy measurements in computer systems
  - jb1236@student.uni-lj.si

Computer Communications Lab

- Research:
  - **Mobile computing**: sensing and machine learning, energy-efficient mobile computing
  - **Security**: online deception, psychology of internet fraud, authentication in IoT environments
  - **Networking and infrastructure**: virtualization, software-defined nets
- BS theses topics and research internships available
- Location: R 3.53, R 3.72



Platform-Based Development

Computing Platforms

- Contemporary computing landscape is extremely diverse

https://www.theedge.co.nz/home/whats-good/2023/06/apples-new-vr-headset-looks-like-its-straight-out-of-a-black-mirror-ep-and-idk-how-to-feel.html

https://www.medicaldesignandoutsourcing.com/developing-software-for-safety-in-medical-robotics/

https://www.indigo9digital.com/blog/how-six-leading-retailers-use-augmented-reality-apps-to-disrupt-the-shopping-experience

Computing Platforms

- Different platforms bring different affordances:
  - High-capability computing
  - Small size
  - Energy efficiency
  - (Wireless) connectivity and internetworking
  - Sensing and actuating
  - High-definition screens
  - Virtual/augmented reality
  - Security
  - … and many others

Computing Platforms

- Different platforms bring different constraints:
  - Intermittent availability
  - Limited processing power
  - Limited storage/memory
  - Lack of (long-range) connectivity
  - Unpredictable execution times
  - Lack of security assurances
  - Difficult, error-prone program
  - … and many others

Computing Platforms

- Different platforms bring different programming paradigms:
  - Low-level programming
    - Get closer to the hardware
    - Assembler, C
  - High-level programming
    - Sometimes don’t even know which machine the programme is running on
    - JavaScript
  - Different operating systems and application programming interfaces (APIs) available

Course Goals

- Examine software development for specific platforms
  - Software engineering perspective

Which programming language should I use for programming X?

What are the limitations of Y platform?

Which platform should I use for achieving Z?

Course Goals

- Many of the platforms you are already (or can get) familiar with through other courses:
  - Web platform development
  - Cloud platform development
  - Game platform development
  - Parallel computing platform programming
- Therefore we focus on:
  - Embedded programming (less)
  - Mobile computing (more)

Course – Embedded Programming Aspects

- Embedded computers represent 98% of all computers in the world
- Increasingly important as the Internet of Things
- Sensing
- Actuating
- Resource constraints
- Arduino programming

https://mazzyautomations.co.za/iot-systems/

Course – Mobile Computing Aspects

- No clear line between embedded and mobile
- Multiple applications
- Personalised use
- Context sensing
- Wireless connectivity
- Android programming
  - Not a basic Android course you will program a lot!

We will concentrate on smartphones

Sensing

Background processing

Distributed applications

Learning Outcomes

Course Outcomes

- After you successfully complete the course, you will be able to:
  - Differentiate between specifics of different computing platforms;
  - Argument selection of specific hardware and software for the development of a product;
  - Understand embedded system development limitations and affordances, and be able to develop a basic embedded system application;

Course Outcomes

- After you successfully complete the course, you will be able to:
  - Describe mobile computing concepts and list different applications of mobile computing;
  - Develop advanced mobile computing applications that involve sensing, user interaction, and wireless data transfer;
  - Analyze and optimize the efficiency of the developed multiplatform computing solution.

Course Components

Lectures

- Help you get a big picture
- Ask for clarifications
- Voice your opinion – discuss!
- Thursdays 3:15pm – 6pm at P20
- There is no comprehensive book for this class!
  - Slides and readings on Ucilnica
  - Take notes!

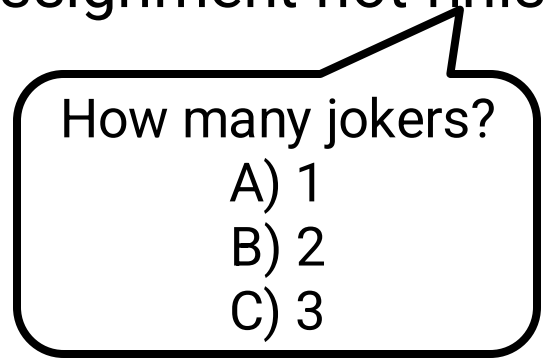
Lab Sessions

- Topical discussions
  - Mandatory readings before the class on ucilnica
  - In-person discussion – participate!
- Arduino programming
  - Assignments done in the lab and in pairs
- Android programming
  - Assignments done in the lab or individually
  - It could be useful to have an Android phone
    - Oreo (8.0) API 26 or higher
  - You need a laptop/PC running Android Studio

You must attend your assigned session!

Lab Sessions

- Discussion lab:
  - Mandatory attendance
- Programming labs:
  - Attend the lab and make meaningful progress
  - Submit the code for grading before the deadline
  - You will be randomly called to discuss your submission
  - Jokers for assignment not finished/defended



Course Project

- Programming your own Android and Arduino application
  - One Arduino part
  - Two Android parts
- Specifications released three weeks before the firm deadline
- Submissions:
  - Bitbucket account
  - Create a private project
    - Add “pbdfrita” as a read-only user
  - Only the final commit will be evaluated

Course Project

- Arduino part specifics:
  - Work in pre-assigned pairs
  - Work during three labs + an extra session
  - Submit the code + very brief report
  - Grading: pass/no-pass/pass with distinction
- Android part specifics:
  - Peer grading
    - Numeric scale
  - Random checks
  - Oral defence of the final part

Course Project

- Peer evaluation:
  - Grading rubric released with the app specification
  - Each student gets to evaluate three other students
  - Assess each assigned app and upload a short report
  - Deadline one week after the programming deadline
    - No report -> no points for your app
  - Final grade is an average of the three grades
- TA evaluation:
  - Apps and reports randomly evaluated by the TA
  - Dishonest reports carry negative points

Course Project

- Oral defence:
  - 5-10mins per student
  - Not understanding your code or the concepts -> fail the project (and likely the course)

Course Project

- To be done strictly individually!
- Support:
  - Slack workspace **pbdfri2025.slack.com**
  - Lab topics are related to the project
  - You must try to fix your code yourself, TAs are not there to debug it

Exam

- Written exam after the lectures
- Related to:
  - Lectures
  - Labs
  - Readings
  - Course project
- Closed book
- Practice exam at the end of the semester
- Conditioned on course project and lab mark!

Policies and grading

Final mark

- Coursework 50%
  - Labs 10%
  - Course project 90%
  - Brief checkpoints 0%
- Exam 50%

Need at least a half of the coursework

Need at least 7 labs

Need at least a half of the exam points

- Mark: ceil[(exam+coursework)/10]
  - If (exam+coursework)/10 is close to an integer, you might be called for a brief oral exam

Policies

- Read the syllabus
- Subscribe to ucilnica and Slack workspace
- Use English for all course-related communication
- No cheating!
  - Do not copy solution
  - Do not allow others to access your work
  - Labs and course projects are to be done individually (except for Arduino part)
  - Cite already published work

Feedback from previous years

"Java is outdated, we learn three new languages in the third year of studies, why not adding Kotlin to it?"

"Android apps should be done in Kotlin, that would make programming simpler"

"I suggest that the next time you don't specify the course project to such a great detail…"

"too much theory"

"Course project specification could be more detailed"

"I liked the peer grading, it's good to see someone else's code, you learn from that"

"Students grade the apps incorrectly"

"the professor is excited about the class…"

"the professor is too excited about the class, expects too much from the students"