ICT Start-Up Programming Curriculum Meeting

September 8 2015

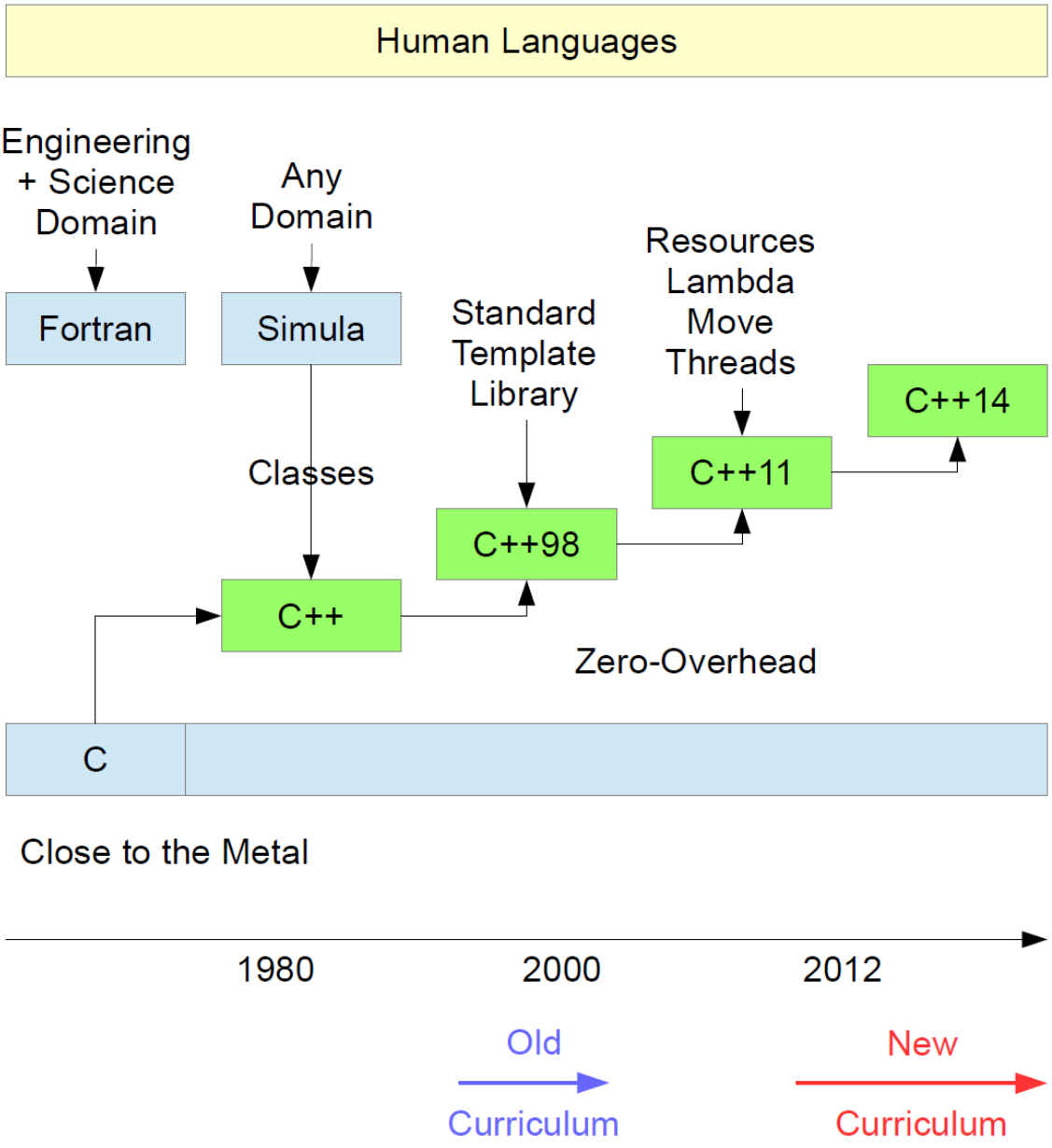
Agenda

1) Overview
    - Philosophy
    - Central Topic Continuity

2) Current State
    - Approved Learning Outcomes
    - Experiments – OOP244 Success
    - Exam and Test Best Practice Suggestions

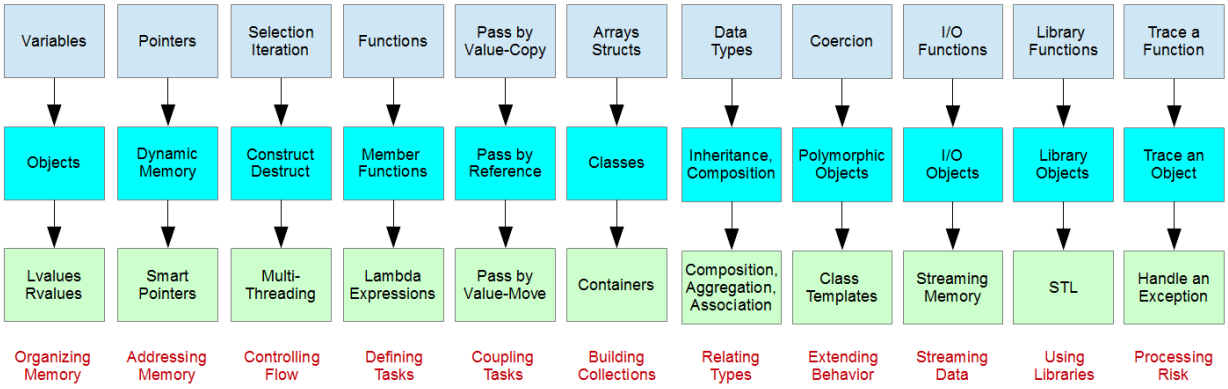
3) Looking Forward

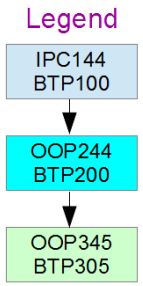
Overview - Philosophy



Overview – Central Topic Continuity

Core Programming Stream - Why is This Topic Important?





Chris Szalwinski
7/9/15

Shifts in Focus

- Primary:
  - Containers, STL, RAII, Smart Pointers, Multi-Threading
- Secondary:
  - Multi-Dimensional Arrays, Linked Lists, Bit Operations

Shifts in Delivery

- Distributed Summative Assessments
  - ~ 10 Quizzes + Test + Exam
- Distributed Formative Practice
  - 10 Workshops on Primary Topics

Current State – Approved Learning Outcomes

IPC144

1 - Design functions using selection and iteration constructs to solve a programming task
2 - Connect functions using pass-by-value and pass-by-address semantics to assemble a complete program
3 - Design collections using arrays and structures to manage data efficiently
4 - Code algorithms using standard library functions to incorporate existing technology
5 - Stream data using standard library functions to interact with users and access persistent text
6 - Trace the execution of a procedural program to validate its correctness
7 - Code complete programs using appropriate object and pointer types to solve programming problems
8 - Explain the purposes of procedural programming features to inform business persons

BTP100

1 - Design functions using selection and iteration constructs to solve a programming task
2 - Connect functions using pass-by-value and pass-by-address semantics to assemble a complete program
3 - Design collections using arrays and structures to manage data efficiently
4 - Code algorithms using standard library functions to incorporate existing technology
5 - Stream data using standard library functions to interact with users and access persistent text
6 - Trace the execution of a procedural program to validate its correctness
7 - Develop algorithms using procedural programming concepts to communicate coding plans
8 - Code complete programs using appropriate object and pointer types to implement specified coding plans
9 - Explain the purposes of procedural programming features to inform business persons

OOP244

1 - Design classes with dynamically allocated resources to model the components of a programming solution
2 - Design member functions using logic constructs to solve tasks of linear complexity
3 - Relate classes using inheritance hierarchies to minimize the duplication of object code
4 - Design polymorphic objects to amplify the reusability of program code
5 - Use stream objects to interact with users and access persistent data
6 - Trace the execution of object-oriented code to validate its correctness
7 - Code a complete program using polymorphic objects to solve a systems or business problem
8 - Explain the purpose of an object-oriented programming feature to inform a business person

BTP200

1 - Design classes with dynamically allocated resources to model the components of a programming solution
2 - Design member functions using logic constructs to solve tasks of linear complexity
3 - Relate classes using inheritance hierarchies to minimize the duplication of object code
4 - Design polymorphic objects to amplify the reusability of program code
5 - Use stream objects to interact with users and access persistent data
6 - Trace the execution of object-oriented code to validate its correctness
~~7 - Code a complete program using polymorphic objects to solve a systems or business problem~~
7 - Code complete programs using polymorphic objects to implement specified coding plans
8 - Explain the purpose of an object-oriented programming feature to inform a business person
9 - Develop an algorithm using object-oriented concepts to solve a simple programming problem

Current State – Experiment – OOP244 Success

Engagement Strategy

- 1 deliverable per week
- 10 Workshops as mini-assignments
- 10 Quizzes
- 1 Common Final Project
- Summative evaluations distributed over Quizzes, Test and Final Exam (50%)

Coordination

- Instructors meet weekly – face-to-face, skype, or big blue button
- Instructors agree timeline
- Instructors prepare workshops and final project – adjust learning outcomes and share workload equally
- Instructors discuss test questions jointly – deliver individual tests and quizzes

Current State – Exam and Test Best Practices

1) provide students with code that is missing a central core and ask them to code that core
2) write a complete unit of code (a short program)
3) provide students with code and ask them to explain it in 3 short sentences
4) ask students to explain in 1 short sentence the purpose of a specific concept (interview questions are important in every semester)
5) test those learning outcomes that are gateway outcomes to the immediately following course(s)
6) print the final exam in color with syntax highlighting
7) provide students with an English description (grammar) and ask them to code the corresponding algorithm
8) limit the weight of the walkthrough(s) to significantly less than 50% of the exam
9) avoid testing all learning outcomes on the final exam - test the most recent one and those not tested summatively throughout the course

Looking Forward

DSA555/BTP500 – needs to be done

- formulate comprehensive set of learning outcomes
- revise the topic list to align with OOP345/BTP305
- consider 2+2 mode of instruction

IPC144 – suggestions to improve retention?

- bringing evaluation weights more in line with OOP244 weights
- weekly coordination meetings
- use both Windows and Linux environments
- proposal to reformat the delivery

Reflecting on Assigned Tasks

- https://sites.google.com/site/reflection4learning/why-reflect (Helen Barrett and Jonathan Richter. University of Oregon, Center for Advanced Technology in Education)
- Evaluate as part of learning outcomes
- The Sink-Hole Paper
    - https://tltl.stanford.edu/sites/default/files/files/documents/publications/2014.JLS-BWPSCK.Programming.pdf
    - lines of code
    - switches from tinkering to planning and vice versa are conducive to learning

OOP345/BTP305 – needs to be done

- clarify learning outcomes to remove some generalities
- introduce 10 quizzes
- move marginal sections in the notes to appendices
- incorporate C++14 features

OOP244 –

- continuing and refining the 10% final exam experiment

Mapping to Core Literacies

- weekly meetings task – map topics to core literacies
- applies to all programming faculty

Call for Volunteers/Contributors

- increase our familiarity with the evolving C and C++ standards
- study and summarize best coding practices w.r.t. our core courses