Exam 1 Format

- Deployed on Canvas, but must take it in the classroom or via DRC arrangement
  - Must sign in – sign-in sheet will be passed along during the exam time
  - Bring Bronco ID or any other form of ID
- Open-book test
  - Use your own words/examples to answer questions
  - Tell your own stories if the question asks for your opinions/experiences/…
  - If you must use/copy someone else's work, quote it " .. " and cite reference.
- Standard time: 70 minutes

Question Format

- Simple questions
  - multiple choices, true/false, fill in blanks, …
- Short answer questions
  - describe a language “feature” in your own words
  - explain a code example using language features
  - illustrate a language feature using code examples (don’t copy online examples)
  - brief justification, pros/cons etc.
    - Use language evaluation criteria and followed by justification
    - That include pros/cons, advantages/disadvantages, supporting which criterion, …

Question Group – multiple version of a question

- Each question is associated with a group of questions
  - Random choice

- You're welcome to study with your group members, however, for the test, even for group language related questions, we're not asking for group consensus, each member should have his/her own opinion in answering a question

- Also, a question may have multiple versions, so you and your study partner may get different version of a question
  - don't compare answers during and after the test
  - don't discuss test problems until after the test has been graded
  - Never release test problems (especially online!)

Review Strategies (for your reference only)

- Read textbook covered chapters/sections
- Study lecture slides
- Go over quizzes and homework problems
- Review your group activity assignments
- Searching your memory for what covered in group sharing
  - Close your eyes, think about what you've benefited from group sharing
  - Make a summary of the important facts covered in Activity 2

Coverage

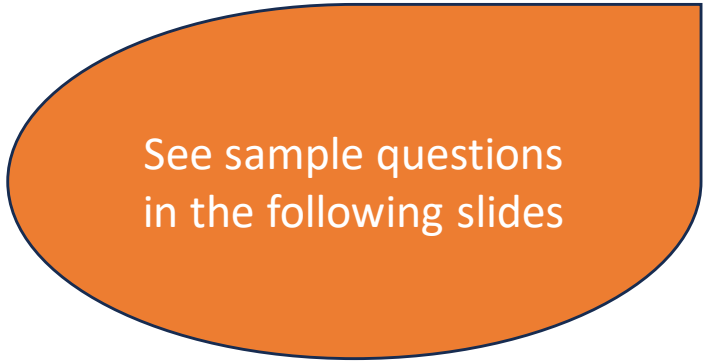
- Lecture 1: Language evolution and evaluation criteria
  - Basic ideas (common sense) on language evolution
  - Evaluation criteria
- Lecture 2: Syntax and semantics
  - Definitions of syntax/semantics, syntax/semantic errors
  - Understanding of EBNF
  - Description of simple codes using operational semantics
- Lecture 3: Names/variables, scoping and lifetime
  - Attributes of variables
  - Lifetime (meaning/definition); 4 categories of variables; 4 categories of arrays
  - Scope rules, nested scopes, static/dynamic scoping

Coverage (cont.)

- Lecture 4: Data Types
  - Data type: type name, range of values, set of operations
  - Mutable and immutable data types
  - Problems with pointers
  - Trends in the design
- Lecture 5: Expressions
  - Various types of expressions (arithmetic, relational, Boolean, conditional, ...)
  - Evaluation orders, short-circuit evaluations
  - Assignments
    - Augmented, increment and decrement assignments
    - multiple and multiple target assignments, …
    - Assignments as expressions

Free-style Questions

- From group sharing
  - General ideas on the trends of language design
  - Your personal impression on what shared by the groups
  - …
- Your group language
  - Need to know the covered language features
  - Doesn’t involve extensive coding
    - Simple code examples may be required



Sample Questions

- The following slides show a few sample questions

- These sample questions are used:
  - to illustrate question style/format only
  - not implying complete coverage

Group Languages

Note: some are open questions, i.e. without standard answers. However, if justification needed, please state clearly.

- Your group language – short answer question.
  - What categories of arrays supported by your language? For each category give a “code” example.
  - In terms of array data type, do you think your language’s support has better/worse readability/writability/reliability than Java? Justify briefly.

- Group Sharing
  - During the presentations of group activities, we got to know some languages don’t have explicitly type declarations.

  True/False: If a language doesn’t have explicitly type declarations, it must be a weakly typed language, i.e. not all type errors could be caught.

  Short answer: implicit type declarations or optional type declarations are getting popular with the newer languages. In your opinion, what are the advantages and disadvantages of using implicit/optional type declaration? Briefly justify your answer.

Lecture 1: Overview and Evaluation Criteria

- Sample question on evaluation criteria
  - E.g. Java enforces array index range checking. With regard to language evaluation criteria, this strategy enhances which of the following? Choose best answer.
    A. Readability
    B. Writability
    C. Reliability
    D. Portability

Lecture 2: Syntax and Semantics

- Review Homework 1

- Java-like language
- Syntax error, semantic error, or no error?

```
int x = 10, y = 20;
while (x < y) {x++; result += y*2; };
```

- Compile-time error, run-time error, or no error

```
int x = -5, y = 2;
do {x++; result += y*2; } while (x+y);
```

Lecture 3: Names, Bindings, Scope, and Lifetime

- Anonymous variables
  - A variable may have name or be anonymous. Identify how many variables (including anonymous variables) are there in the following Java-like code?

    double f;

    String s = new String (“hi”);

    A: 1 B: 2 C: 3 (including anonymous object) D: 4

- 4 categories of variables
  - Static, stack dynamic, explicit heap dynamic, implicit heap dynamic
  - (note: 4 categories of arrays: static, fixed stack dynamic, fixed heap dynamic, heap dynamic; note: some language uses stack dynamic – see sharing.)

C++ like code:

double f (int x, int y) { …. }

x and y belong to which category of variable?

A. static

B. stack dynamic

C. explicit heap dynamic

Lecture 4: Data Types

- Homework 2
- Array category
  - Java-like array

    ```
    int[] a = new int[5];
    ```

    which category of array it belongs to?

    A. static
    B. stack dynamic
    C. fixed stack dynamic
    D. fixed heap dynamic
    E. Implicit heap dynamic

- Pointer problems

  C++ like code, what problem may arise from the following code?

```
int *p;
p = new int[10];
…
p = new int[100];
```

  A. Memory leak (uncollected garbage space)

  B. null pointer

  C. Unsafe access of unallocated memory location via pointer

Sample short answer questions

- Java array: homogeneous array
- Python list: heterogeneous array

- What is your view in supporting arrays in programming languages? State pros and cons and briefly justify your answer.
  1) Should a language support both homogeneous and heterogeneous array types (i.e. one data type for homogeneous array and another data type for heterogeneous array)
  2) just homogeneous arrays
  3) just heterogeneous arrays

Lecture 5: Expressions and Assignments

True or False: Operator precedence and associativity rules are defined mathematically, thus Ruby and Java have the same precedence and associativity rules.

Simple or short answer questions: Given the following Java-like code (note: code may be incomplete),

```
f(): a function
        b = true;           //b a Boolean variable
        b = b || d();
        return b;
d(): a function
        displays “d() executed”;
        return true;
```

Question: When f() is called, will d() executed be displayed? Why or why not?

Sample Short answer question:

What is short circuit evaluation?

What are advantages of short circuit evaluation?

Write a code example that involve short circuit evaluation and use this example to explain the advantage of using short circuit evaluation?

Give a code example that illustrates augmented assignment.

What are the advantages of using augmented assignment? (name 2)

End of review.

Questions, discussions?