Midterm

Started: Jun 27 at 3:05pm

Quiz Instructions

During this midterm test, you may access

1. Notes that you have taken during class (handwritten *or* electronic)
2. Study aides that *you* (or your study partners) have created (e.g., quizlet, evernote, studyblue, goconqr, tinycards, etc)
3. The textbook
4. The Daily PLs (including anything linked therefrom)
5. Class code (available at **https://github.com/hawkinsw/cs3003** (https://github.com/hawkinsw/cs3003))
6. Slides (original and annotated)

I trust that you will *not use the Internet to answer questions on this exam other than to access the materials above*. By submitting your exam responses through Canvas, you assert that you have not broken this trust. Violation of this accord will result in immediate and severe consequences.

With that out of the way, I know that you will do great on this test! Be sure to,

1. Read the questions closely,
2. Respond to *all* the aspects of the essay questions, and
3. Check your work.

If you have any questions during the test, please come see me! If you are in doubt about something, please ask! It's likely other students are having the same doubts!

There are 110 total points available but the exam is worth 100. In other words, 10 points of extra credit are available.

Good luck -- you won't need it!

Question 1 — 2 pts

A(n) ________________ *(two words)* programming language is one where the type/variable binding is done before the code is run and does not change throughout program execution.

Question 2 — 2 pts

A(n) ________ occurs when the language implicitly converts a variable of one type to another.

Question 3

2 pts

A(n) ____________ *(two words)* is a pattern of problem-solving thought that underlies a particular genre of programs and languages.

Question 4

2 pts

A language with ________________________________ (*two words*) is one without side effects.

Question 5

2 pts

In a(n) ___________________________________ programming language, the *assignment* statement (or expression) is akin to the pipeline between the memory and the processor in the Von Neumann Model.

Question 6

2 pts

A dynamically typed language *cannot* be strongly typed.

○ True

○ False

Question 7 2 pts

The _________________________ *(two words)* of a statement contains all the name/variable bindings visible at that statement.

Question 8 2 pts

A change between types done at the explicit request of the programmer is known as a(n) _______________.

Question 9 2 pts

A variable's ______________________ is the period of time during program execution when it is associated with computer memory.

Question 10 2 pts

A(n) ________________ is an association between an entity and an attribute.

Question 11
2 pts

______ is/are the rules for constructing a structurally valid computer programs in a programming language.

Question 12
2 pts

__________________ define the effect of each statement's execution on the program's operation.

Question 13
2 pts

_______________ is the range of statements that can access a name binding.

Question 14
2 pts

In ________________________ polymorphism, the subprogram's semantics are always the same.

Question 15

2 pts

A(n) ________________ is a subprogram that does not return any value(s).

Question 16

2 pts

In a computer based upon the ____________________ (*two [or three] words*) model, instructions and data are stored together in the same memory.

Question 17

2 pts

In a language with __________________________________ (*two words, first hyphenated*), evaluation of a Boolean expression is halted as soon as the result is completely determined.

Question 18

2 pts

_________________________________ is the process of removing the details to simplify and focus attention on the essence. Some others like to say that _____________________ (*same word, no need to type it twice*) is all about remembering what's important and forgetting the rest (depending on the context, of course!).

Question 19

2 pts

________ functions do not have side effects.

Question 20

2 pts

In a(n) ______________________ scoped language, the scope of a variable can be determined by analysis of the program's source code.

Question 21

4 pts

Which of the following are *true statements* about subprograms? (Select all that apply)

- ☐ All subprograms are expressions.
- ☐ Only one subprogram is active at a time.
- ☐ A subprogram has multiple entry points.
- ☐ Program execution returns to the caller upon completion.

Question 22

4 pts

Which of the following is/are attributes of a variable? (Select all that apply)

- ☐ Scope
- ☐ Access
- ☐ Type
- ☐ Representation
- ☐ Name
- ☐ Primitive

Question 23

4 pts

Which of the following are the valid lifetimes for a variable.

- ☐ Modern
- ☐ Classic
- ☐ Explicit heap dynamic
- ☐ Stack dynamic
- ☐ Runtime
- ☐ Static
- ☐ Implicit heap dynamic

Question 24

4 pts

Which of the following are considered by *Sebesta* to be valid binding times? (Select all that apply)

- ☐ Language design
- ☐ Program execution
- ☐ Language implementation
- ☐ Program debugging

☐ Program design

Question 25

4 pts

In a *statically typed programming language*, all the variables must have a type at the time of compilation. Please select *all* the different ways that a variable can be given a type at the time of compilation. (Again, please check *all* the ways!)

☐ Implicit - Convention

☐ Scoped

☐ Abstraction

☐ Implicit - Inference

☐ Assumed

☐ Explicit

Question 26

5 pts

Consider this snippet of code (that *is **not** Python but shares a similar syntax!):*

```
def function_a():
    variable_v = "outer variable"
    def function_b():
        value_of_variable_v = variable_v
        print(f"variable_v: {value_of_variable_v}")

    def function_c():
      function_b()

    def function_d():
      variable_v = "not the outer variable."
      function_b()

    function_d()

def main():
    function_a()
```

In this *hypothetical* language,

1. Every function has its own scope (named the same as the function),
2. **scoping is dynamic**, and
3. program execution begins with the `main` function.

When this program is invoked, execution begins at the `main` function which invokes `function_a`. In turn, `function_a` invokes `function_d`. `function_d` invokes `function_b`. Order the scopes that will be searched (from innermost to outermost) for `variable_v` when it is accessed in `function_b`.

1	[Choose]
2	[Choose]
3	[Choose]
4	[Choose]

Question 27

15 pts

You are designing a programming language that combines a runtime stack and the use of stack-dynamic storage for local variables to make it possible for programmers to write recursive functions. Describe how these two features interact so that a person writing an application with your language can use recursion.

In your answer, include responses to the following topics and/or questions:

1. What is the runtime stack?
2. Why recursion cannot be used when local variables are have a static lifetime?
3. At what point(s) in a program is the runtime stack modified?
4. What is the performance impact of using stack-dynamic storage versus static storage for local variables?
5. What is the space efficiency of using stack-dynamic storage versus static storage for local variables?

Edit View Insert Format Tools Table

12pt Paragraph

p 0 words

Question 28

15 pts

In class we discussed Algebraic Data Types (ADTs). ADTs come in two different flavors: sum types and product types. Please describe each of the two types. For each of the types you must

1. Choose a particular language and describe if/how that type is actually implemented. For the language that you choose, consider each of the following questions in your response:
    1. Does the language implement the sum/product type perfectly or must there be a *combination* of types that work together to accomplish the implementation (e.g., you add a *tag* to a data structure that the language supports natively)?
    2. In your opinion, does that language make it easy or hard to work with its version of the sum/product type? In other words, are the operations on sum/product types that we discussed in class easily expressed in the language?
2. Explain the reason that the Sum/Product type were given their names.
3. Describe a use-case that is particularly well suited for use by the Sum/Product type. In other words, describe a particular case where the availability of Sum/Product types makes code more expressive than it would be in the absence of such types.

Edit View Insert Format Tools Table

12pt Paragraph

p

0 words

Question 29

15 pts

Short-circuit evaluation plays a very important role in the semantics of the evaluation of Boolean expressions in a programming language. The user of a programming language needs to know if/how their language uses short-circuit evaluation. But, why?

Please describe the way that the presence/absence of short-circuit evaluation in a programming language can effect the behavior/correctness of a program written in a given language. To receive full points for the answer,

1. define short-circuit evaluation;
2. describe why designers might include short-circuit evaluation in their language;
3. explain how short-circuit evaluation can/does interact with other parts of the language's semantics to affect program behavior/correctness. Don't forget to address the characteristics of a programming language that would make it so that short-circuit evaluation *cannot* change the outcome of a program's behavior/correctness.

If it helps, feel free to refer of the differences in the potential behavior of the following two code (***pseudocode***) snippets in your answer:

Snippet A

```
# A function that does something and then returns
# true or false.
def sfn() -> bool:
    ...

# A function that does something else and then
# returns true or false.
def rfn() -> bool:
    ...

def main() -> None:
    if rfn() and sfn():
        ...
    if rfn() or sfn():
        ...
```

Snippet B:

```python
# A function that does something and then returns
# true or false.
def sfn() -> bool:
    ...

# A function that does something else and then
# returns true or false.
def rfn() -> bool:
    ...

def main() -> None:
    if sfn() and rfn():
        ...
    if sfn() or rfn():
        pass
```

Edit View Insert Format Tools Table

12pt Paragraph

p 0 words

Quiz saved at 3:05pm Submit Quiz