CMSC 245 Principles of Programming Languages
Fall 2020
Exam 1

This exam contains 10 Questions on pages numbered 1-13. This exam is designed to be taken in 80 minutes.

Please fill in all answers in the space provided in the form. Be sure to save your filled out exam as **First-nameLast-name.pdf**. The exam should be e-mailed to your Professor no later than 1:00p Eastern Time on Thursday, October 7, 2020. No credit will be given for exams submitted late. Your Professor will confirm the receipt of the exam by e-mail.

All resources (text book, class notes, completed labs and assignments, etc.) are permitted, but no assistance from another person. It is okay to email the Professor if you require a clarification.

Good Luck!

Declaration

Sign the following statement **after** you have completed the examination by typing your name in the box provided. Your exam will **not be graded** without your signature:

I certify that my responses in this examination are solely the product of my own work and that I have fully abided by the Bryn Mawr College Academic Integrity policy and instructions stated above while taking this exam.

Name: ______________________

For grading purposes only. Go on to next page.

Question	**Points**	**Max Points**
1		**10**
2		**10**
3		**20**
4		**10**
5		**10**
6		**20**
7		**10**
8		**15**
9		**10**
10		**10**
Total		**125**

Question 1 (10 points) In the space provided, categorize the following programming languages as either **imperative** or **functional**:

1. C
2. C++
3. C#
4. Haskell
5. Javascript
6. CommonLisp
7. Python
8. Rust
9. Smalltalk
10. Swift

Question 2 (10 points) For each of the following programming languages specify (Yes/No) whether they support object-oriented programming

1. C
2. C++
3. C#
4. Haskell
5. Javascript
6. Lisp
7. Python
8. Rust
9. Smalltalk
10. Swift

Question 3 (20 points) In the space provided, write down the technical term used that each of the following describes.

1. Refers to the connection between the CPU and memory that used to access instructions as well as data.
2. Translates a source code program into equivalent machine-level instructions.
3. Sews together the program with the libraries it uses to form a complete executable machine code program.
4. Relocates an executable program into specific locations in memory just prior to running the program.
5. An interpreter uses this to take a statement in a program and decode and execute it interactively.
6. Converts a program written an assembly language into an equivalent machine language program.
7. The mechanism used to keep track of function calls and their referencing environment.
8. Region of a program where a binding is active.
9. Functions with no name.
10. Complete set of bindings at a given point in a program.

Question 3. *contd.*

11. When one or more names in a program refer to the same object at the same point in a program.

12. When the scope of all names can be known by looking at the text of the program. I.e. at compile time.

13. When the scope of a name can only be determined at run time, dictated by flow of execution of program.

14. Place in memory where dynamically allocated data is stored.

15. A local variable in a function that remembers its value across function calls.

16. The data structure (or mechanism) used to keep track of function calls during program execution.

17. Runtime mechanism used to reclaim unused memory in a heap.

18. Associating a name in a program with its attributes.

19. The data structure used to keep track of all bindings during compilation.

20. The mechanism used to convert primitive values into reference objects in Java.

Question 4 (5+5 points) What are first-class objects in a programming language?

For each of the following specify whether they are or not first-class objects in Java. Write a Yes/No in the space provided.

int	
double	
boolean	
array	
function	
class	

Question 5 (10 points) What makes a programming language object-oriented? Be specific and describe in no more than 2-4 sentences. Mention three object-oriented programming languages.

Question 6 (20 points) Consider the skeletal program below:

```
1 program Q5 {
2      char n;
3      int x = 2;
4      function W() {
5            int y;
6            println(n);
7      } // W()
8
9      function D() {
10            char n;
11            int z;
12            n = ‘D’;
13            W();
14       } // D()
15
16       for (int i = 0; i < x; i++) {
17             println(i);
18       }
19       n = ‘Q’;
20       W();
21       D();
22 } // Q5
```

Part 1 (5 points): Name all the symbols defined in the program above. Use the line numbers provided to indicate the location of their definition.

Question 6. *contd*.

Part 2 (5 points): For each symbol that is a variable, identify whether it is a *local*, *non-local*, *global variable*. Again, if needed, use line numbers to identify their location. If a symbol is local identify its scope by naming the body of the program unit (you can also use a range of line numbers (e.g. the program from line 2-22, etc.).

Part 3 (5 points): If the programming language uses *static scoping rules*, write the output(s) from the program above?

Part 4 (5 points): If the programming language uses *dynamic scoping rules*, write the output(s) from the program above?

Question 7 (10 points) Here is an algorithm for computing the GCD of two numbers **a** and **b**:

```
while (a ≠ b)
    if a > b
        a ← a – b
    else
        b ← b - a
```

Assume that **a** and **b** are defined as integers and bound to values greater than 0.

Part 1 (5 points): Code the above steps in Java:
(**NOTE:** just the steps above)

Part 2 (5 points): Code the above steps in Go:

Question 8 (15 points) Consider the steps below to compute the minimum of two variables **a** and **b**:

```
if (a < b) then
      min ← a
else
      min ← b
```

Part 1 (5 points) Show how the steps above can be coded using the *ternary conditional operator* in Java:

Part 2 (5 points) Describe the *ternary conditional operator* in the Go programming language. Based on your answer, show how the above steps would be coded in Go:

Question 8. *contd*.

Part 3 (5 points) Of the programming languages listed below, write a Yes/No whether the programming language supports the **?:** operator.

1. C	
2. C++	
3. C#	
4. CommonLisp	
5. Fortran	
6. Haskell	
7. Java	
8. Javascript	
9. Rust	
10. Swift	

Question 9 (10 points) Define *value model* and *reference model* for variables in a programming language. Which model does Java use?

Question 10 (10 points) What are *immutable objects*? Given an example of an immutable object in Java.