

CS 3360 - Design and Implementation of Programming Languages

HOMEWORK 3: DATA TYPES AND OOP
(File \$Date: 2018/10/29 21:34:16 \$)

Due: November 6, 2018

This homework may be done individually or in pair.

The purpose of this homework is (a) to know various data types supported by most programming languages and their design issues and (b) to understand support for object-oriented programming, in particular in Java and C++.

1. (5 points) What are advantages and disadvantages of decimal data types?

2. (10 points) Write a short discussion of what was lost and what was gained in Java's designers' decision to not include the pointers of C++.

3. (10 points) What are the arguments for and against Java's implicit heap storage recovery, when compared with the explicit heap storage recovery required in C++?

4. (10 points) Multidimensional arrays can be stored in row major order, as in C++, or in column major order, as in Fortran. Develop the access functions for both of these arrangements for three-dimensional arrays.

5. (10 points) Make two lists of applications of matrices, one for those that require jagged matrices and one for those that require rectangular matrices. Now argue whether just jagged, just rectangular, or both should be included in a programming language.

6. (10 points) Explain the advantages or disadvantages of having all values in a language be objects.

7. (10 points) Summarize the fundamental argument for dynamic method binding. Why do C++ and C# use static method binding by default?

8. (5 points) What is one programming situation where multiple inheritance has a significant advantage over interfaces?

9. (5 points) Explain why allowing a class to implement multiple interfaces in Java and C# does not create the same problems that multiple inheritance in C++ creates.

10. (10 points) Describe the issue of how closely the parameters of an overriding method must match those of the method it overrides. Consider the rule of Java 8.

11. (15 points) Rewrite the following C++ classes in Java and compare the result with the C++ version in terms of readability and writability. Hint: Both `stack_2` and `queue_2` are `*private*`

subclasses.

```
class single_linked_list {
private:
    class node {
    public:
        node* link;
        int contents;
    };
    node *head;
public:
    single_linked_list() {head=0};
    void insert_at_head(int);
    void insert_at_tail(int);
    int remove_at_head();
    int empty();
};

class stack_2 : private single_linked_list {
public:
    stack_2(){}
    void push(int value) {
        single_linked_list :: insert_at_head(value);
    }
    int pop() {
        return single_linked_list :: remove_at_head();
    }
    single_linked_list :: empty;
};

class queue_2 : private single_linked_list {
public:
    queue_2(){}
    void enqueue(int value) {
        single_linked_list :: insert_at_tail(value);
    }
    int dequeue() {
        single_linked_list :: remove_at_head();
    }
    single_linked_list :: empty;
};
```

`};`

12. (10 bonus points) Explain subtyping for Java 5 Generic (parameterized) classes? I.e., when an instantiated class is a subtype of another instantiated class and why?

WHEN AND HOW TO TURN IN

Turn in your solutions at the start of class on the due date. No

late submissions will be accepted unless arrangements have been

made in advance or unless unusual circumstances warrant an

exception.

GRADING

Clarity is important; if your writings are sloppy and hard to read,
you will lose points.