Marco Seman

Professor Jie Shen

CIS 400 Programming Language Fall 2018

December 1, 2018

Assignment #3

Table of Contents

[Question 1 (10 points) 1](#_Toc531000613)

[Question 2 (10 points) 2](#_Toc531000614)

[Question 3 (10 points) 3](#_Toc531000615)

[Question 4 (10 points) 3](#_Toc531000616)

[Question 5 (10 points) 4](#_Toc531000617)

[Question 6 (10 points) 5](#_Toc531000618)

[Question 7 (10 points) 5](#_Toc531000619)

[Question 8 (10 points) 5](#_Toc531000620)

[Question 9 (10 points, bonus question) 6](#_Toc531000621)

[Question 10 (30 points). 6](#_Toc531000622)

[Question 11 C++ Interview (37 points; 1 point for each sub-question). 6](#_Toc531000623)

[Question 12 C# Interview (47 points; 1 point for each sub-question). 15](#_Toc531000624)

[Question 13 Java Interview (45 points; 1 point for each sub-question). 24](#_Toc531000625)

# Question 1 (10 points)

Write a Scheme function that takes one parameter, a simple list of numbers, and returns the sum of all these numbers

i.e., (sum ‘(2 3)) 🡪 5.0

(sum ‘(-1.5 2)) 🡪 0.5

**Scheme Q1.**

(define (sum nums)

(if

(null? nums)

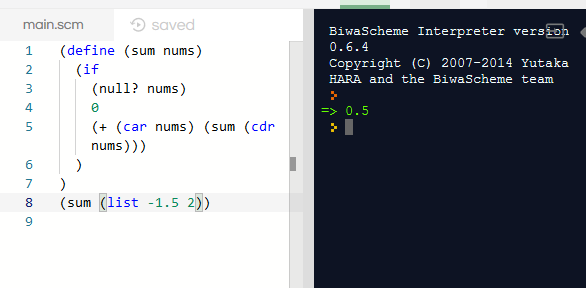
0

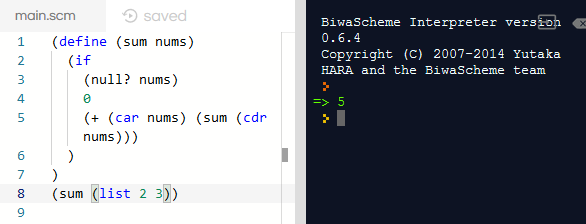
(+ (car nums) (sum (cdr nums)))

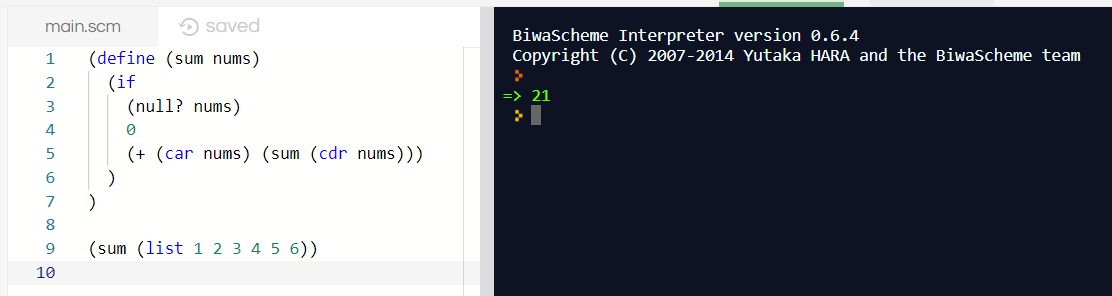
)

)

(sum (list -1.5 2))







# Question 2 (10 points)

Write a Scheme function that takes one parameter, a simple list of numbers, and returns the average of all these numbers.

i.e., (simpleaverage ‘(1 2 3)) 🡪 2

(simpleaverage ‘(0.5 -1.2 0.7)) 🡪 0.0

**Scheme Q2.**

(define (sum nums)

(if

(null? nums)

0

(+ (car nums) (sum (cdr nums)))

)

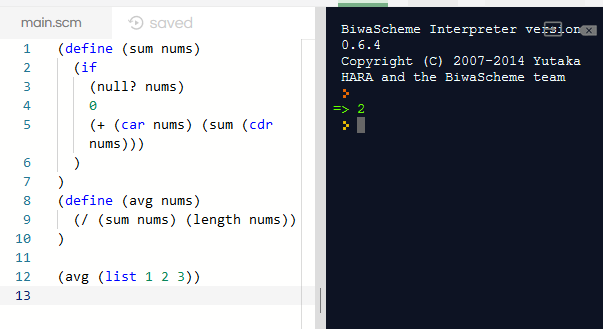
)

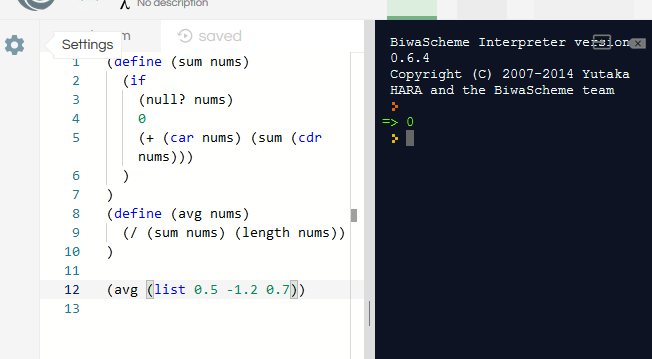
(define (avg nums)

(/ (sum nums) (length nums))

)

(avg (list 1 2 3))





# Question 3 (10 points)

Write a Scheme function that takes one parameter, a possible nested list of numbers, and returns the average of this nested list of numbers.

i.e., (nestedaverage ‘(1 2 (2.5 3 3.5))) 🡪 2.4

(nestedaverage ‘((0.3 0.5 0.7) -1.2 0.7)) 🡪 0.2

**Scheme Q3.**

(define (flat nums)

(cond ((null? nums) '())

((pair? nums) (append (flat (car nums))

(flat (cdr nums))))

(else (list nums))))

(define (sum nums)

(if

(null? nums)

0

(+ (car nums) (sum (cdr nums)))

)

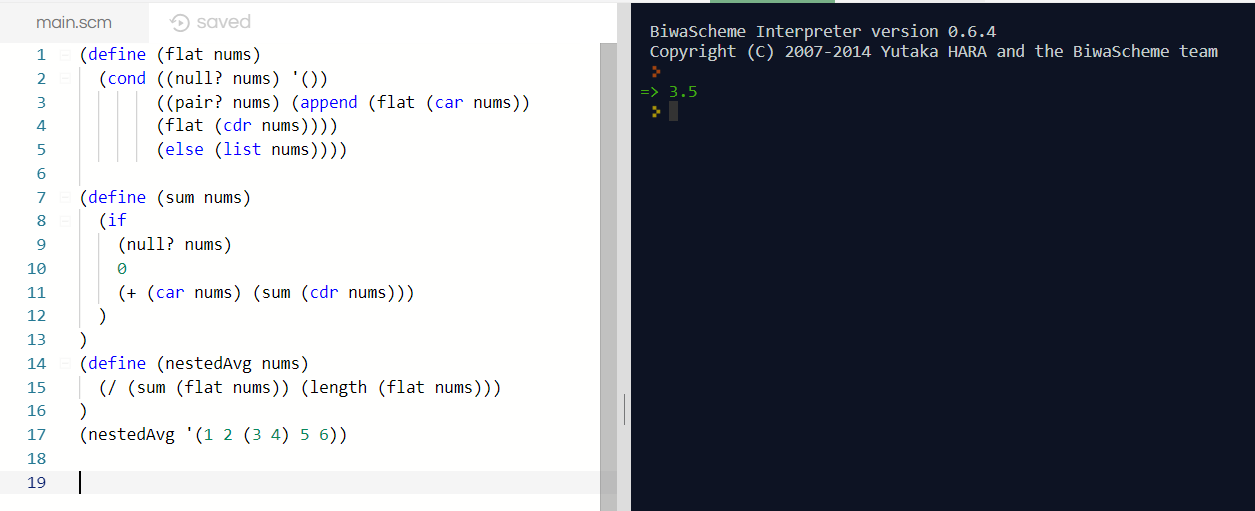
)

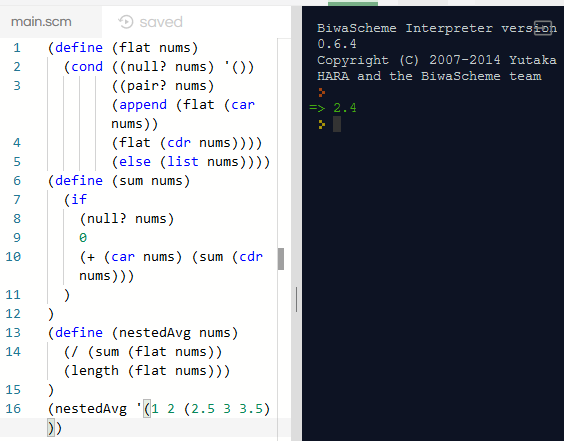
(define (nestedAvg nums)

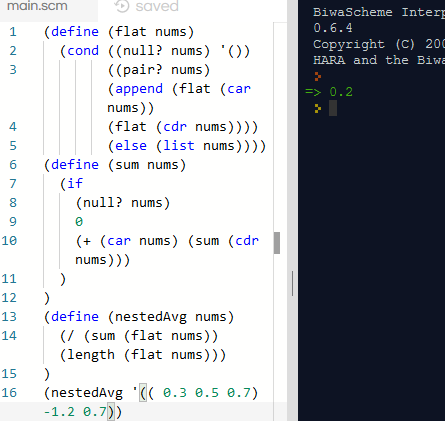
(/ (sum (flat nums)) (length (flat nums)))

)

(nestedAvg '(1 2 (3 4) 5 6))







# Question 4 (10 points)

Write a Scheme function that returns number of sublists in a list.

i.e. (numsublist ‘(1 2 3)) 🡪 0

(numsublist ‘(1.2 2.1 3.7 (2 3))) 🡪 1

**Scheme Q4.**

(define (sublst nums)

(cond

((null? nums) '0)

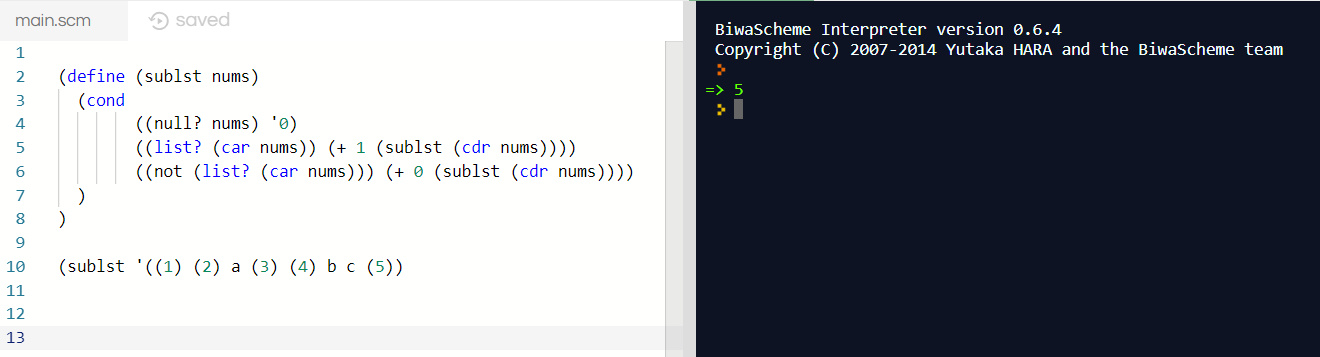
((list? (car nums)) (+ 1 (sublst (cdr nums))))

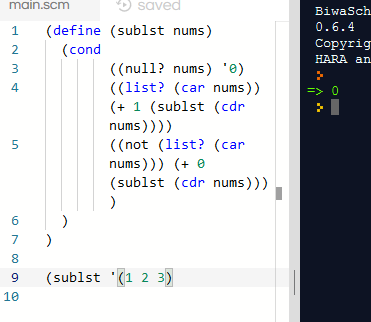
((not (list? (car nums))) (+ 0 (sublst (cdr nums))))

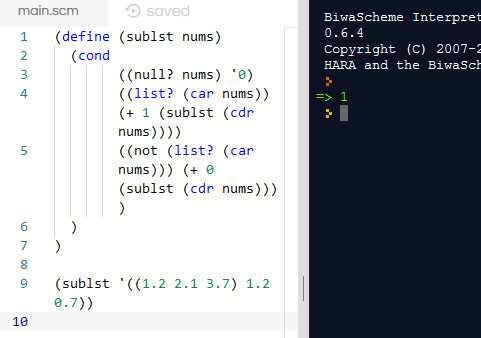
)

)

(sublst '((1) (2) a (3) (4) b c (5))







# Question 5 (10 points)

Write a Scheme function that takes a list as a parameter and return a list with the last top-level element removed. If the given list does not have anything, the function should return ().

i.e., (remove ‘(1)) 🡪 ( )

(remove ‘(1 2)) 🡪 ( 1 )

(remove ‘(1 2 3)) 🡪 (1 2)

(remove ‘(1 2 3 (4 5))) 🡪 (1 2 3)

* If you use xlisp, you can store each function into a text file and load into lisp environment by typing:

(load “text file name”)

then you can test your functions.

* You can download xlisp.exe and xlisp.dll from xlisp.org. But, you are suggested to use Dr Scheme (<http://racket-lang.org/>) because it has a nice GUI.

**Scheme Q5.**

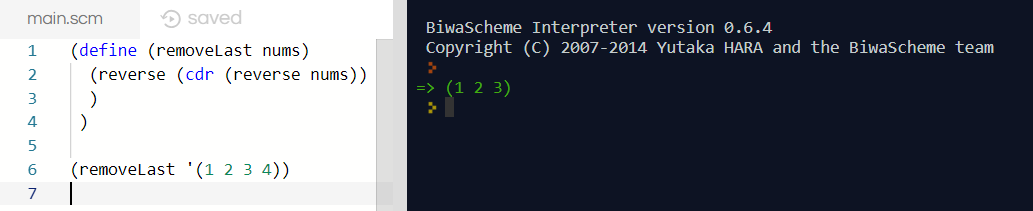
(define (removeLast nums)

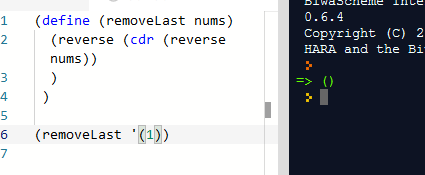
(reverse (cdr (reverse nums))

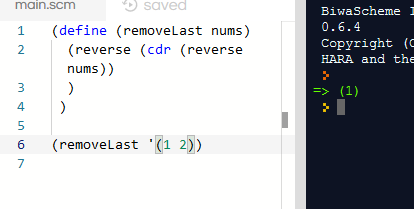
)

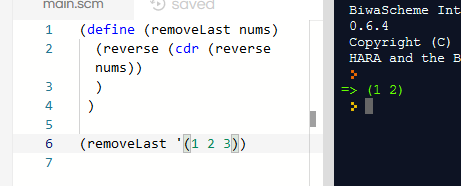
)

(removeLast '(1 2 3 4))









# Question 6 (10 points)

Write a Prolog function with two parameters, an input list and an output variable that is the summation of the absolute values of all numbers (including 0) in the input list.

i.e.,

?- sum([1,2,3],X).

X = 6

Yes

?- sum([1,2,-3,-4],X).

X = 10

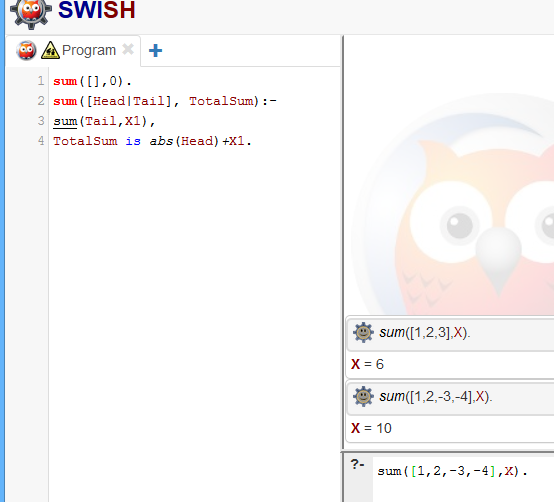
Yes

sum([],0).

sum([Head|Tail], TotalSum):-

sum(Tail,X1),

TotalSum is abs(Head)+X1.



# Question 7 (10 points)

Write a Prolog function that takes two parameters: an input list and an output variable that is the smallest one among the squares of each number in the input list.

i.e.,

?- min([3, 5], X).

X = 9

Yes

?- min([3.5, 5.5, 0], X).

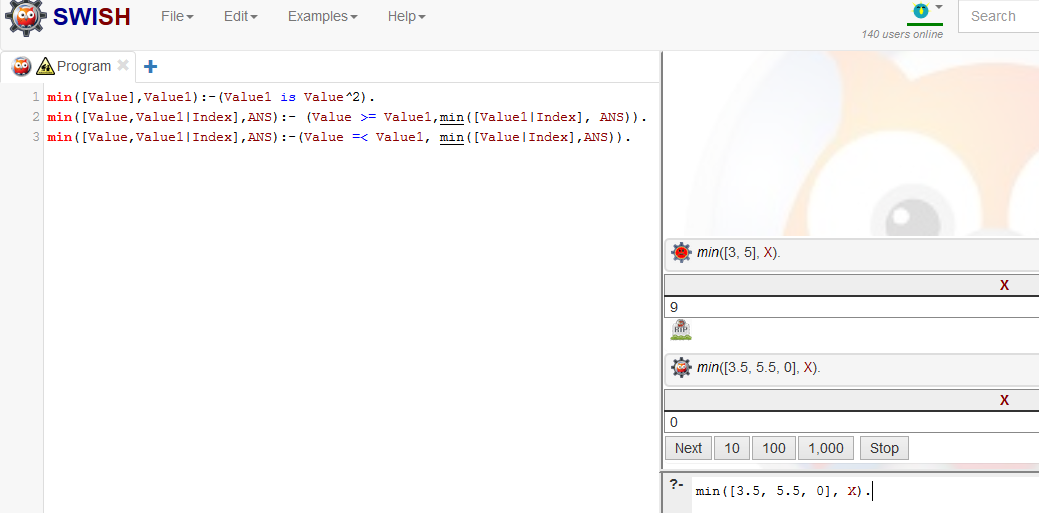
X = 0

Yes

min([Value],Value1):-(Value1 is Value^2).

min([Value,Value1|Index],ANS):- (Value >= Value1,min([Value1|Index], ANS)).

min([Value,Value1|Index],ANS):-(Value =< Value1, min([Value|Index],ANS)).



# Question 8 (10 points)

Write a Prolog function that takes three parameters: two input lists and an output variable that is equal to 0 if the length of the first list is less than that of the second list, and becomes 2 if the length of the first list is greater than that of the second list. If the lengths of the two lists are equal, then if the sum of all numbers in the first list is less than or equal to that of the second list, the output value equals 1. Otherwise, the output value becomes 4.

i.e.,

?- out\_list([15, 3.5, 5.5], [1.0, 2.0], Y).

Y = 2

Yes

?- out\_list([1, 2, 3], [2, 3, 4, 5], X).

X = 0

Yes

?-

Code:

out\_list(Head, Tail, X) :-

(length(Head,Length),length(Tail,Length2),

Length<Length2, X is 0).

out\_list(Head,Tail,X):-

(length(Head,Length),length(Tail,Length2),

Length>Length2, X is 2 ).

out\_list(Head,Tail,X):-

(length(Head,Length),length(Tail,Length2),

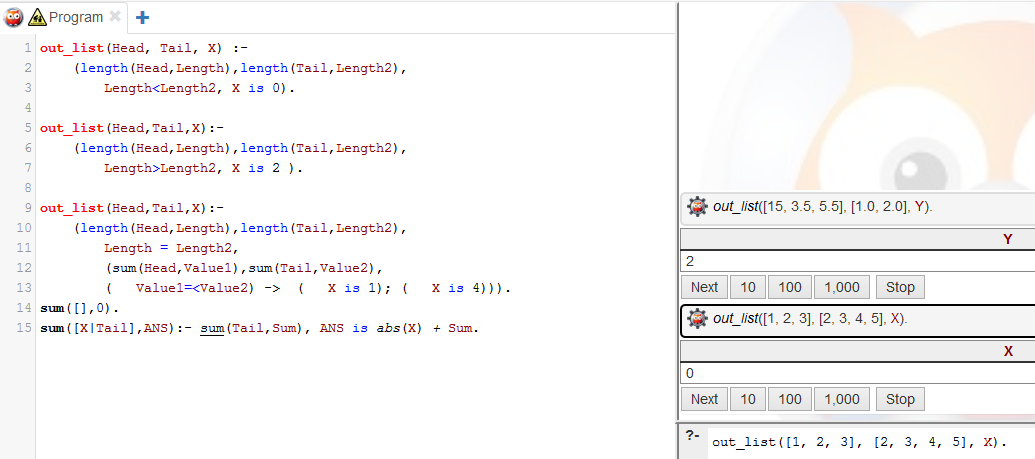
Length = Length2,

(sum(Head,Value1),sum(Tail,Value2),

( Value1=<Value2) -> ( X is 1); ( X is 4))).

sum([],0).

sum([X|Tail],ANS):- sum(Tail,Sum), ANS is abs(X) + Sum.



# Question 9 (10 points, bonus question)

Baron, Cook, Frank, Mike, and Sam live in a five-story building. Baron doesn’t live on the 5th floor and Cook doesn’t live on the first. Frank doesn’t live on the top or the bottom floor, and he is not on a floor adjacent to Sam or Cook. Mike lives on some floor above Cook. Who lives on what floor for all the five people ? Use prolog to solve this problem.

Hint: ‘\=’ is the operator of ‘not equal’. ‘= : =’is the operator to evaluate the values of two variables and then compare them for equality.

level([level(\_,5), level(\_,4), level(\_,3), level(\_,2), level(\_,1)]).

structure(Stories) :- level(Stories), person(level(Baron, B), Stories), B \= 5,

person(level(Cook, C), Stories), C \= 1, person(level(Frank, F), Stories), F \= 1, F \= 5,

person(level(Mike, M), Stories), M > C, person(level(Sam, S), Stories),

not(adjacent(S, F)),not(adjacent(F, C)), print\_story(Stories).

print\_story([A | B]) :- write(A), nl, print\_story(B). print\_story([]).

individual(X, [X | \_]). individual(X, [\_ | Y]) :- person(X, Y).

individual(X, Y) :- X =:= Y+1. individual(X, Y) :- X =:= Y-1. not(Goal) :- \+ call(Goal).

?-structure(X)

# Question 10 (30 points).

Polymorphism is one cornerstone of Object-Oriented Programming. This question has the following setting of a polymorphism problem:

#include "pch.h"

#include <iostream>

using namespace std;

class Animal {

    int legs;

public:

    Animal() {legs = 4;}

    virtual void talk() { cout << "Animal Can't Talk!\n"; }

};

class Cow : public Animal {

public:

    void talk() { cout << "Moo!\n"; }

};

class Pig : public Animal {

public:

    void talk() { cout << "Grunt!\n"; }

};

class Snake : public Animal { };

int main()

{

    Animal Arr[4];

    Cow cow;

    Pig pig;

    Snake snake;

    Arr[1] = cow;

    Arr[2] = pig;

    Arr[3] = snake;

    Arr[0].talk();

    Arr[1].talk();

    Arr[2].talk();

    Arr[3].talk();

}

1. Output:

Animal Can’t Talk!

Animal Can’t Talk!

Animal Can’t Talk!

Animal Can’t Talk!

1. Output:

Animal Can’t Talk!

Moo!

Grunt!

Animal Can’t Talk!

1. Output:

Animal Can’t Talk!

Moo!

Grunt!

Animal Can’t Talk!

1. C# and Java both have dynamic binding. So, whatever is in the class of that array element is what is printed by the talk function. C++ does not have dynamic binding. So, whatever is in the base class ends up being what is printed by the talk() function.

# Question 11 C++ Interview (37 points; 1 point for each sub-question).

1. How do you decide which integer type to use?

Deciding the integer type is depended on the requirement. When we are required to use an value that uses two bytes from with value ranges from -32768, +32767 we use short int. for values that are positive and use values from 0 to 65535 we used unsigned int. if four byes are needed, we use regular int or long int, for unsigned int, we use four bytes that are all positive. We use long long int if we want to use 8 bytes.

1. What does extern mean in a function declaration?

Extern functions can be accessed outside the score of a .cpp file. A variable, function, or declaration is defined with extern that allows us to make sure the usage f variable of the function remains part of the current source file. \* In other words, Extern means that the object or function is accessible through its name from other translation unit in the program.

1. What’s the auto keyword good for?

An auto keyword is good for declaring an object with an automatic storage duration. The auto function takes the object and destroys it at the end of the objects scope. Since the default for local variables is auto, you don’t need to manually specify it in some languages.

1. Define a linked list node which contains a pointer to itself.

Declaring a basic linked list that contains a pointer to itself:

Struct node {

Int info;

Struct node \*next; }

Int main() {

// create a new node

Struct node \*node = (struct node \*) malloc(sizeof(struct node));

Return 0; }

The propose of the malloc function is that it allocates memory located in the heap section. The function then returns a void pointer that’s pointing to the first byte in memory.

In other words:

Linked lists link to the value we use to point to the next node. To point to anything we need a pointer. Here is where we make the relationship between linked list and pointer: linked list starts a data structure where it points to the first node, then continues and points to the next node. Linked list is a data structure where we link different blocks of data when dynamically allocating the address.

1. How do I declare an array of N pointers to functions returning pointers to functions returning pointers to characters?

char \*(\*(\*a[P])())();

Using typedefs make the code more readable

typedef char\* (\*functiontype\_one)(void);  
typedef functiontype\_one (\*functiontype\_two)(void);  
functiontype\_two myarray[N];

1. How can I declare a function that returns a pointer to a function of its own type?

Use typedef to implement it:

typedef int (\*f)(int);

typedef f (\*g)(float);

typedef g (\*h)(char);

1. What can I safely assume about the initial values of variables which are not explicitly initialized?

Variables that are aren’t initialized the compiler can assign any garbage value to the uninitialized variable.

1. How can I create a two-dimensional dynamic array ?

A 2D array is a array of pointers to an array

A normal array in C++ looks like:

int\* array = new int[Size]

A Dynamic 2D array looks like:

Int\*\* array = new int\*[row];

For(int I =0; I < row; ++I)

Array[I] = new int[column];

If row was 4 and column was 3 the 2D array would look like:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Array[0] 🡪 | Array[0][0] | Array[0][1] | Array[0][2] | Array[0][3] | Array[0][4] |
| Array[1] 🡪 | Array[1][0] | Array[1][1] | Array[1][2] | Array[1][3] | Array[1][4] |
| Array[2] 🡪 | Array[2][0] | Array[2][1] | Array[2][2] | Array[2][3] | Array[2][4] |

1. How can I deallocate a two-dimensional dynamic array ?

for (int i = 0; i < row; i++) {

for (int j = 0; j < column; j++) {

Array[i][j] = NULL;

}

}

delete [] Array;

Array = NULL;

1. What is a virtual function ?

A virtual function is a member of a function that is declared within a base class and is overridden by its derived class. To refer to a derived class object, we use a pointer or a reference to the base class. Virtual functions ensure that the correct function is called for an object regardless of the type of reference used for function call.

1. How do I do dynamic binding ?

Reference to a member function that is resolved at runtime

Class Base{

Public:

Virtual void Operation();

}

Class DerivedBase: Public Base{

Public:

Void Operatoin()

}

Main{

Base \* pBase = new DerivedBase()

pBase🡪Operations()

}

1. When should I use a local static variable in a function ?

Local static variables are used when the value of the variable is independent of the object, but not unique for each object.

1. When should I use “register” to declare a variable ?

The registry variable should be used for heavily used variable like loop control to improve performance by decreasing the access time. Another use of register is when you use pointer variables where the register can have access of a memory location. Using an operating a register variable

1. Can I put a function into a struct ?

In C, no. You however do it in C++. A struct class is a class with all members being public by default in the language.

1. How do I initialize a pointer to a function?

Different ways to initialize a pointer:

Int \*ptr1=NULL;

Float \*ptr2=NULL;

Char \*ptr3=NULL;

Double \*ptr4=NULL;

We’ll use Int \*ptr1=NULL; in this scenario.

To declare a function:

Int foo(int)

To declare a pointer to a function:

Int \* foo(int)

1. Are arrays row-major or column-major ?

Arrays are row major since their implementation is more so like this:

|1, 2, 3|

|4, 5, 6|

|7, 8, 9|

Instead of this:

|1, 4, 7|

|2, 5, 8|

|3, 6, 9|

1. What is the difference between a normal pointer and a void pointer?

A void pointer is pointer that points to some data location in the storage where it doesn’t have any specific value.

A normal pointer is a pointer that points to an object in a certain location with nothing else.

1. What will the following program do?  
   void main()  
   {  
   int i;  
   char a[]="String";  
   char \*p="New Sring";  
   char \*Temp;  
   Temp=a;  
   a=malloc(strlen(p) + 1);  
   strcpy(a,p); //Line number:9//  
   p = malloc(strlen(Temp) + 1);  
   strcpy(p,Temp);  
   printf("(%s, %s)",a,p);  
   free(p);  
   free(a);  
   } //Line number 15//  
       
   a) Swap contents of p & a and print:(New string, string)  
   b) Generate compilation error in line number 8  
   c) Generate compilation error in line number 5  
   d) Generate compilation error in line number 7  
   e) Generate compilation error in line number 1

B) Error on line one where it is supposed to be int, not void since main must always return an int. Once changing it to int, the error is then on line 8 since you cant add an integer to a char value.

1. enum number { a= -1, b= 4,c,d,e};  
   What is the value of e ?  
   (a) 7  
   (b) 4  
   (c) 5  
   (d) 15  
   (e) 3

the value of e would be: a) 7

because the value reads b = 4 and adds 1 afterwards to the next value. c = 5, d = 6, e = 7

1. What is the difference between overloading and overriding ?

Overloading – the ability for functions with the same name to de defined if the methods have different signatures.

Overriding – the ability of an inherited class to be rewritten of the virtual method of the base class.

1. What is the difference between private and protected access ?

Private – Only accessible within the same class it’s declared in. A member with no access modifier is only accessible within classes in the same package.

Protected – Accessible within all classes in the same package and within the subclasses in other packages.

1. When are copy constructors called?

Copy constructors are called when instantiation one object and initializing it with values from another object. When the object tis returned from a function by value.

1. What is difference between malloc()/free() and new/delete?

Both of these allocate and release memory, however, they have several different features.

Malloc/Free – memory allocated from heap, returns a void pointer, returns null on failure, must specify size required in bytes, allocating array requires manual calculation of space, reallocating larger chucks of memory simple, they will not call new/delete, cannot be overridden legally, no way to splice user code into the allocation sequence to help with low memory.

New/Delete – memory allocated from free store, return’s a fully typed pointer, new standard version never returns a null, are called with type id, has a version to explicitly handle arrays, operator new/delete can be overridden legally, can add new memory allocator to deal with low memory, constructor/destructor is used to initialize/destroy object.

1. How do I overload an assignment operator ?

First we declare new resources to allocate new memory and copy the values

Class Array {

Int \* ary

Int value;

Public:

Array& operator=(const Array& other)

{

If(this!=&object) {

Int\* newAry = new int[object.value];

Copy(other.ary, other.ary + other.value, newAry)

Then we remove old resources:

Delete[] ary;

Then we allocate the new resources to handle the object:

Ary = Array;

Value = other.value;

And we must always “return \*this;” at the end of the assignment operator.

1. What is special to static data members and member functions in a class ?

Static member function – can be called even without any object of the class that exists.

The Static functions are only accessed using the class name and scope resolution operator ::. The static member function can only access static data member and other static member function of any functions from outside the class.

Static data members – is declared with the static keyword and is known as class member, thus only a single copy if the variable creates for all objects.

1. what is the difference between "new" and "operator new" ?

New – starts with operator new to allocate memory, then invokes the constructor for the right type of object resulting in a object being created in memory. Any changes to one member function reflects on all other objects in member function.

Operator new – only allocates raw memory, nothing else.

1. What is the purpose of keyword “volatile” ?

Volatile is intended to prevent the compiler from applying any optimizations on objects that can change in ways that cannot be determined by the compiler. Once an object has been declared as volatile, they are omitted from optimization due to their values being changed by the code outside the scope of the entire source code.

1. What is the purpose of keyword “mutable” ?

Mutable is used when there is a variable inside a class that is used within that class to signal things like mulex or a lock. The variable doesn’t change the behavior of the class, but its necessary in order to implement a thread safety of the class itself.

1. What is pure virtual function? or what is abstract class?

Pure Virtual function – a virtual function that’s declaration ends in = 0 at the end of function declaration. A class that contains a pure virtual function is abstract meaning you cannot create an object of the class.

Abstract class – An abstract class are classes where it contains more than one abstract methods. The abstract method is a declared method without any implementation. Abstract classes require subclasses to provide implementation for abstract methods.

1. Can a copy constructor accept an object of the same class as parameter, instead of reference of the object?

No - A copy constructor cannot accept an object of the same parameter. The proram will generate an error with the first object of the reference variable.

1. What is a local class? Why can it be useful?

Local class are defined in a block of groups of zeros or several statements between braces. Local classes are defined in the body of a method. Local classes are useful when you need to create one instance of a class, access its constructor, or introduce a new named type.

1. What is a nested class? Why can it be useful?

A nested class is a class that is defined in another class.

Reasons nested classes are useful:

* Increases encapsulation
* Logically group classes to be grouped in one place
* Can be more readable and maintainable
* They can be visually illustrated in terms of child class to parent class connection

1. How do you access the static member of a class?

// Declaring a static member of a class

Class A

{

Public:

// create a public instance that returns valueA

Static int getValue()

{ return valueA; }

Private:

Static int valueA;

};

// initialize the value. this works even though valueA is private since it’s a definition.

Int A::Value = 1;

Int main()

{

// to access the content that’s in A

Cout << A::getValue();

}

In other words, if we wanted to access a private static value, we must create a public return function to be able to obtain the information in main().

1. What does extern "C" int func(int \*, Foo) accomplish?

The command will turn off name mangling for function so that one can link the code compiled by the c compiler.

1. Can you overload a function based only on whether a parameter is a value or a reference?

Yes, you can overload a function based on only the parameter value or a reference. It is fine to have both a parameter or a reference to coexist since they are different.

1. What happens when you make call "delete this;" ?

When calling “delete this;” it will call the destructor of the object it was called on and send the allocated memory back to the heap. In other words, delete this releases the memory to the memory manger resulting in the machine to no longer be able to make any guarantees about the content of what is in the memory. The pointers called will still be valid, but the machine will no longer guarantee the contents of the memory being valid.

1. What is inline function?

Inline function – C++ enhancement feature to increase the execution time of a program. The function instructs the compiler to take the inlines so that the compiler can replace the function definition to wherever they are called. A compiler replaces the definition to the compiler time rather than referring the function definition at runtime.

# Question 12 C# Interview (47 points; 1 point for each sub-question).

1. Can multiple catch blocks be executed?

Yes, multiple catch blocks can be executed with try statements. You can start with catching specific exceptions and then in the last block, you can use the catch base exception.

1. What is Jagged Arrays?

A Jagged array is an array of arrays. Each array element of an can be a different size or dimension.

1. What is the difference between ref & out parameters?

Ref – A parameter that has a value before it goes into a function. The called function can read or change the value at any time. In other words, the parameter goes in then comes out.

Out parameters – A parameter with no official value before going into a function and the function must be initialized. In other words, the parameter only goes out.

1. What is the use of using statement in C#?

The using statement is useful for objects where their lifetimes are within the method or block in where they are created. It is useful to reduce the code where resource-insensitive operations are performed by similar syntax and removing multiple calls to release resource at the end of the scope.

1. Can “this” be used within a static method?

No, because a static method doesn’t have an instance of a class. “this” is used when you want to point to an instance of a class.

1. What is special to static constructor ?

A static constructor is special for initializing any static field associated with a type of operation. It is also useful for reading required configuration data into read only fields. The static constructor is automatically run at runtime the first time its needed.

1. What is difference between constants and readonly?

Constants – are static by default, the must have a value at compilation time, cant declare within functions, are copied into every assembly that uses them, can be used in attributes.

ReadOnly – must have a set value by the time the constructor exits and are evaluated when instance is created.

1. What is an interface class?

An interface class is like an abstract base class. A class or struct that implements the interface must implement all its members. An interface cannot be instantiated directly to its members where they are implemented by any class or struct that implements the interface.

1. What are value types and reference types?

Value type – data type that colds a data value in its own memory space. The variable of the data types directly contain their value such as: int i = 100;

Reference type – does not store values directly. Reference types store the address to where the value is located.

1. What are sealed classes in C#?

Sealed classes are used to stop classes from being inherited. You also cannot derive or extend any classes from it. The main purpose of sealed classes is to withdraw the inheritance attribute form the user so they cannot attain a class from a sealed class.

1. What is the difference between Array and Arraylist?

Arraylist – grows dynamically

Arrays – Size remains static throughout the program

1. Can a private virtual method be overridden?

No, you cannot override a private virtual method because there isn’t a point since there would be no way to override them. A protected virtual method can be overridden.

1. Describe the accessibility modifier “protected internal”.

Protected internal member is accessed from any type within its contained assembly. Internal member is also accessible in a derived class in another assembly where the access occurs through a variable of the derived class type.

1. What are the differences between System.String and

System.Text.StringBuilder classes?

System String – immutable, therefore you cannot change the size of the strings object

System Test String Builder – mutable where we can change the size of the stringbuilder class object

1. How can we sort the elements of the array in descending

order?

int[] array = new int[] { 5, 3, 1, 2, 4 };

Array.Sort<int>(array,

new Comparison<int>(

(i1, i2) => i2.CompareTo(i1)

));

1. Write down the C# syntax to catch exception?

Int(main)

{

Char Myarray[10]

Try

{

For(int I =0; I <= 0; i++)

{

If( I > 9) throw “out of range”

Myarray[i]=’z’

} throw to

Catch(char \*\*)

{

Cout << “error”

}

Catch(int x)

{

….

}

1. What are generics in C#.NET?

Generics allows the user to define a class with placeholders for its fields, methods, parameters. Generics also replaces the placeholders with a specific type at compiler time.

1. What is an object pool in .NET?

Object pools can improve application performance in situations where you require multiple instances of a class and the class is expensive to create or destroy. When a client program requests a new object, the object pool first attempts to provide one that has already been created and returned to the pool. If none is available, only then is a new object created.

1. List down the commonly used types of exceptions in .Net?

ArgumentException, ArgumentNullException , ArgumentOutOfRangeException, ArithmeticException, DivideByZeroException ,OverflowException , IndexOutOfRangeException ,InvalidCastException ,InvalidOperationException , IOEndOfStreamException , NullReferenceException , OutOfMemoryException , StackOverflowException etc.

1. What are Custom Exceptions?

Defined classes where all exceptions are inherited from a base system.exception class.

1. Why can’t you specify the accessibility modifier for methods inside the interface?

Because all interface methods are public and you cannot create an access modifier in an instance. If you want to use one, then you use abstract class.

1. How can we set class to be inherited, but prevent the method from being overridden?

You must use private access specified to prevent the class from being overridden in the derived class. The keyword sealed can be used to the methods to prevent further overriding’s of the methods.

1. Is Struct value-type or reference type ?

Structs are value types while the classes are reference types.

1. What happens if the inherited interfaces have conflicting method names?

There may be an issue pertaining to the methods from different interfaces except for the different data. The compiler will however allow it.

1. How to use nullable types in .Net?

To declare nullable types to variables in .Net.

Nullable<int> i = null;

We use Nullable<T> where T declare the data type. The nullable type represents the precise range of values associated to the data type. Nullable<int> can be assigned to any numeric value between + and – 214748364.

1. How we can create an array with nondefault values?

string[,] array = null;

array[0, 0] = new string("0x0".ToCharArray());

array[0, 1] = new string("0x1".ToCharArray());

array[1, 0] = new string("1x0".ToCharArray());

array[1, 1] = new string("1x1".ToCharArray());

array[1, 2] = new string("1x2".ToCharArray());

array[1, 3] = new string("1x3".ToCharArray());

1. What is difference between is and as operators in c#?

Is – checks if an object can be cast to a specific type

As – attempts to cast an object to a specific type, and returns null if it fails

1. What’s a multicast delegate?

A multicast delegate in C# is a function pointer that allows the programmer to take an encapsulated reference to a method inside of a delegated object.

The delegated object is then passed to the code where the method is referenced without having to know the compile time where the method is invoked.

1. What are indexers in C# .NET?

An indexer allows an object to be indexed like an array does. For a class, when defining an indexer, the class behaves similar to a virtual array where you can access the instance of the class using the array operator [].

1. Is C# code is managed or unmanaged code?

C# is managed code since it is not compiled to machine code but to an intermediate language that is interpreted and executed by some service machine.

1. What does int? mean? Explain the relationship with Nullable.

Int? is a syntactic sugar for the type nullable<T> where T is a int struct. There is no difference between the two, int? is short for nullable.

1. What's the difference between Hashtable and Dictionary<>?

Dictionary – class is a strong type where you must specify the data types for key and value, there is no need for boxing/unboxing, when you try to access non existing key dictionary, your receive an error, dictionary maintains an order of the stored values, there isn’t a need of unboxing/boxing, so hashtable is faster.

Hashtable – non-generic type, is a weakly typed data structure where you can add keys and values of any object type, values need to have boxing/unboxing, when trying to access non existing key hashtable, it gives null values, hashtables never maintains an order of the stored values.

1. Explain IDisposable and the 'using' statement.

IDisposable – an interface with a single method dispose() for removing any unmanaged recources such as files, streams, database connection. The using statement provides the syntax to specify the scope of the use of a resource object and are useful for objects whose lifetimes are within the encapsulation they’re created in.

1. Is System.String a class or a struct (or reference or value type if you prefer).

System string is a class (Reference type) defined in the namespace system.

1. How would you define an Anonymous Functions? What is the difference between Anonymous Methods and Lambda Expressions?

Anonymous functions are functions that were declared without any named identifiers to refer to it. Anonymous functions are also not accessible after its initial creation.

Anonymous methods – allow parameter list to omitted entirely not used within the body and it doesn’t lead to ambiguity.

Lambda Expression – allows interfaces in parameters, can be converted to delegates or expression trees, allow parameter list to be shorted to the parameter name when type is inferred and there is only a single parameter.

1. How does an Enumerator (IEnumerable & IEnumerator) work ? How does an Iterator work in C# 2.0 ?

Enumerator:

An enumerator is a set of integer constants declared using the enum keyword. The enumerator contains its own values that cannot be inherited or passed.

An example of enum:

Enum DayOfWeek {sun, mon, tue, wed, thu, fri, sat };

Sunday would be 0, Monday is 1, Tuesday is 2, and so on.

Iterator:

The iterator tells the compiler to generate the IEnumerable interfaces for the user. Without iterators, the programmer would need to create a class that implements IEnumerable that includes Reset, Current, MoveNext. In other words, iterator calls a generator function which is a specified class of coroutine.

1. What is Garbage Collection ? How does it work ? What are generations in GC ?

Garbage collection manages allocation and reclaim of memory. Garbage collection works on managed heap – a block of memory that stores objects. Generations in GC store and handle long and short lived objects of heap.

1. What is the role of an Extension Method ? When would you use it ?

Extension method – allows you to add methods to existing types without creating new derived type, recompiling, or modifying the original type.

When to use it – when you want to add a method to an existing type without modifying the original source code or using inheritance.

1. What is an event ?

Event – enables a class or object where it notifies other classes or objects when something of interest occurs. The class send the event called the publisher where the class will receive the even called subscribers.

1. How would you define a new custom event ?

Provide an access to the object definition then modify it to declare the custom event

Public event EventHandler<EventArgs> ModelChanged;

Another way to do it is in a private method to invoke the event:

Private void OnModelChange(EventArgs e)

{

If(ModelChanged !=null)

ModelChanged(this,e);

}

1. How would you choose between a pure abstract base class and an interface.

Pure Abstract Base Class:

* Provides more structure
* A class may extend to one abstract class
* Can provide complete code, default code or stubs that have to be overridden.
* Usually defines
* Fast
* Must be rewritten if a third party class is going to use it.

In other words, Abstract classes are useful when you plan on having multiple inheritances. Abstract classes are faster but less dynamic if you are going to implement them in a third party code.

Interface:

* Allows someone to start from scratch
* May implement several instances
* Can change the interface easily by fixing the interface class.
* Others can implement your interface in their code easily
* Slow and requires more indirection
* Disadvantage is that every method in the interface must be public exposing everything.

In other words, you would choose an interface if you need to take the code and implement it easily to another program.

1. Name 3 features you dislike about c#. how would you make them different? What features do you like about C# and why?

Dislike- C# lacks cross platform where it only works for windows and unix but cant build anything for ios. If order to make them different, you would have to reinvent the wheel of the operating system. C# lets you use pointers for unsafe blocks. C# is slower due to its .NET framework.

Like – Integrates well with Windows without any additional special features to get it to work properly. If your system were to get hacked, the hacker wouldn’t automatically have access to the source code. Hackers have to crack the software before seeing any special components. I also enjoy the face that it is interface based in visual studios and you can drag and drop the widgets you like and code their functionalities.

1. Why are catch all exception handlers bad? When/Why might you use them?

Catch all exception handlers are bad because the programmer is supposed to handle the exception properly. A programmer cannot expect the handle all kinds of exceptions in the code.

Catch all exceptions should only be used if the programmer can properly handle them.

1. Does .NET support multiple inheritance?

No, .NET does not support multiple inheritance classes because you have to use interfaces or combination of one class and interfaces where the interfaces should be followed with a class name in the signature.

1. If we have a person class, how can we implement sorting on different properties of the class. (I answered implementing IComparable and IComparer....)

CompareTo method implementation in the person class would be useful. The method will return a string. In CompareTo, we want to sort the names alphabetically, if the names are equal, we will want to check for age or last name and sort alphabetically from there.

IComparer would be useful since it allows sorting an object when implemented. When IComparer is declared, we have to add a public method CompareTo to implement the custom sorting for a class.

1. What is a partial class ?

Partial class – Each class resides in a separate file with the .cs extension. The cs file can have a single class where where it implements multiple cs file using the partial modifier keyword. Partial classes are useful when working on a large project to spread out the amount of code in a sinlge cs file so multiple programmers can program at once.

1. Can a nested class be the derived class of the outer class ?

Nested class do not need to be inherited by anything. Nested class can have access to everything in the contained class such as private fields, properties, or methods that are inside of a public class. Inheriting from a parent class doesn’t allow a nested class to see if its parents private members and methods. However, inheriting from a private class allows a nested class to see its members and methods in if its parent is a protected or public

# Question 13 Java Interview (45 points; 1 point for each sub-question).

1. Is “abc” a primitive value?

No. there are 8 main primitive data types: char, byte, int, float, double, short, Boolean, long

1. What restrictions are placed on the values of each case of a switch statement?

Each case in the switch statement must be evaluated as a value that can be promoted to an int value during compilation time.

1. What is the purpose of finalization?

Finalization is used to perform an action before an object is cleaned up. During a cleanup, the method removes any orphan objects. The purpose is to give an object that isn’t reachable a chance to perform a cleanup process before it is garbage collected.

1. What is garbage collection? What is the process that is responsible for doing that in java?

A garbage collector removes any objects that aren’t being used in the java application anymore. The java virtual machine is responsible for identifying garbage collection in java.

1. Can an Interface be final?

No, an interface cannot be final because we cannot instantiate interfaces. In order to make interfaces useful, we must make subclasses. Final does not allow a class to be extended.

1. Can there be an abstract class with no abstract methods in it?

Yes, there can be an abstract class with no abstract method since abstract classes are independent concepts. When declaring an abstract class, it means a subclass has to provide an implementation for that method.

1. Can an Interface have an inner class?

Yes, an interface can have a nested class or a class within a class.

1. Can we define private and protected modifiers for variables in interfaces?

You can, but it doesn’t make sense to do it conceptually. An interface is similar to a blueprint to a class where the programmer declares methods and classes to implement an interface that is responsible for that definition.

1. What modifiers are allowed for methods in an Interface?

Public and abstract modifiers are allowed for method in an interface.

1. What is Serializable?

Converts object to a byte stream so it can be reverted back into a copy of the object.

1. What is Externalizable?

An interface that extends serializable and adds two methods writeexternal() readexternal() which are automatically called serialization or deserialization.

Deserialization is the reverse process where a byte stream is used to reallocate actual java object to memory.

1. What is a local, member and a class variable?

Local variable – variable where its scope is defined inside a function

Member variable – Operators and functions declared in a class.

Class variable – Specifies attributes or property oof a class that is may be reffered to as a member variable or static member variable.

1. What is the catch or declare rule for method declarations?

For method declaration, the catch or declare rule is a try catch statement. The checked exception is thrown in the body of the method where the method must catch the exception or declare it in the throws clause.

1. What is a native method?

Native Method – indicates that a method is implemented in platform dependent code. In other words, a native method allows the programmer to code from other language such as C++ in your java code.

1. What modifiers may be used with an inner class that is a member of an outer class?

A inner class may use declared as private, protected, public, static, final, or abstract with an inner class that is a member of an outer class.

1. What are the states associated in the thread?

New 🡪 Runnable 🡪 Running 🡪 Waiting/Blocked 🡪 Dead

1. What is synchronization and why is it important?

Synchronization – ability to control access of multiple threads to a shared resource. In other words, it is the process of having a thread execute another thread.

Importance – To improve the consistency of the data that synchronization uses.

1. What are synchronized methods and synchronized statements?

Synchronized Methods – prevents thread interference and memory errors. If an object was visible, to more that one thread, it reads or writes to the objects variable which is completed through synchronized methods.

Synchronized Statements – Similar to synchronized methods where it automatically takes the intrinsic look for the statement of the object and releases it when the method is returned. Statements are more extendable than synchronized methods.

1. What restrictions are placed on the location of a package statement within a source code file?

You must initialize the package statement at the first line in your source code or else there will be a runtime error. There must also be a public class declaration in that file.

1. What are wrapped classes?

Wrapper classes – a class where the object contains one of the 8 primitive data types. It takes the primitive data type and converts it into an object. A wrapped class can also convert an object into a primitive data type.

Types of wrappers: Boolean, Character, Byte, Short, Integer, Long, Float, Double

Example:

int value = 10;

Integer I = Integer.valueOf(value); // this converts an int into a primitive Wrapper

1. What is the eligibility for an object to get cloned?

For an object to be cloned it must pass these scenarios:

1. What are the different identifier states of a Thread?

There are four different identifier states which are:

New: Thread is in a new state when you create an instance of the thread class before the invocation of the start() method

Runnable: Thread in running state after start() method is initialized.

Non-runnable: When the thread is still alive but isn’t eligible to run.

Terminated: when the run() method exits.

1. What kind of thread is the Garbage collector thread ?

Garbage collector thread is a Daemon thread.

Daemon thread: service providers of the thread that are running in the same process as daelmon thread. The thread supports background tasks when normal threads are executed.

1. What is a daemon thread ?

Daemon thread – low priority thread that runs at irregular intervals in the background during compilation time and takes care of the garbage collector operation.

1. What is mutable object and immutable object ?

Mutable object – fields that can be changed after construction.

Immutable object – have no fields that can be changed when an object is created. In other words, an object cannot be altered. For example: String str = “abc”; str = str +”efg” cannot work.

1. What is the basic difference between string and stringbuffer object ?

String – immutable (a string that cannot be altered). In other words, if your string isn’t going to change, then use string.

StringBuffer – mutable (a string that can be altered). Inn other words if you want a string to change and to access multiple threads, then Stringbuffer is useful since it is synchronous for thread-safety.

1. What is the purpose of Void class ?

Void – create a class that will not return a value after the program is executed.

The purpose of Void classes is reflection where we are able to get the return type of a method as void.

1. What is the base class for Error and Exception ?

The base class for error and exception is Throwaway.

Throwaway

/ \

Exception Error

1. What is a package ?

A namespace that organizes that organizes a set of classes and interfaces. In other words, a package is like folders inside of our filing system. When declaring a package, we can take other classes in other files and use them in the source code that you’re working on for readability purposes. It divides the lines of code on one source file and places each class on its own file.

1. Difference between HashMap and HashTable?

HashMap:

* Allows one null key and any number of null values.
* Not synchronized, making it better for non-threaded applications.
* New class introduced in JDK 1.2
* Fast
* Traversed by iterator
* Inherits AbstractMap class
* Iterator is fail fast

HashTable:

* Does not allow null keys or values.
* Synchronized internally, cannot be unsychronized
* Traversed by enumerator and iterator
* Not fail-fast enumerator
* Inherits dictionary class
* Legacy class
* Slow

1. Difference between Vector and ArrayList?

ArrayList:

* Not synchronized
* Increases 50% of current array size if number of elements exceed form capacity
* Fast since it isn’t synchronized
* Uses iterator interface to traverse elements
* Not a legacy class

Vector:

* Are Synchronized
* Increments 100%, double’s array size if total number exceeds more than capacity.
* Legacy class
* Slow since its synchronized
* Uses enumerator interface to transverse elements

1. Difference between Swing and Awt?

Swing:

* Part of Java foundation class
* Components are platform independent
* Components are lightweight since swing sits on top of awt
* Works faster
* Provide more flexible user interface
* Occupies less memory space
* Requires javax.swing
* Widgets are meaningless pixels within a window

Awt:

* Called abstract window tool
* Components are platform dependent
* Components are heavyweight components
* Slower than swing
* Takes more memory than swing
* Requires javax.awt
* Is a cross platform interface
* Uses native GUI widgets

1. What is an Iterator?

A collection framework to retrieve elements one by one. It is used for looping several collections of classes such as hashmap, arraylist, and linked list. The iterator took the place of enumeration which was used to iterate vectors.

1. What are Transient and Volatile Modifiers?-

Transient:

* Keyword used with instances to exclude them from the serialization process.
* Cannot be used with static
* Variable are initialized with default values during deserialization

Volatile Modifiers:

* Keyword used to mark java variables that are being stored in the main memory area.
* Can be used with static
* Every variable is read from the computers main memory, not from the CPU cache.

1. What is the difference between this() and super()?-

Super() – calls the super class constructor

This() – calls constructors in the same class. Calls parameterized constructors.

1. What are inner class and anonymous class?-)

Inner class:

* Are classes that are declared within a class.

Anonymous class:

* Inner classes that are declared without a proper name.
* A way to override a method of a class or an interface.

1. What is the difference between Integer and int?

Integer:

* A wrapper class for an int.
* Wraps primitive types into a real object
* To make it easier to work with, it uses boxing and unboxing to wrap and unwrap the primitive which can be written as: list.add(integer.valueOf(#));

Int:

* One of the 8 primitive data types. It is a 32 bit value that can be used as a return value, parameter, a field, or a local variable type.
* Not an object, has no methods, cannot be included directly in more complex types
* Written as int x = (a number between -2,147,483,648 and 2,147,683,647)

1. What is the difference between Array and vector?-

Array – is a fixed size. Easy to sort. Appropriate for storing fixed number of elements.

Vector – Size can change. Multiple objects are allowed to be stored. Elements can be removed from the vector.

1. What is the difference between exception and error?-

Error – problems with the applications that the compiler should not try to catch

Exception – conditions that are reasonable in the applications that you might want to catch.

1. What is multithreading and what are the methods for inter-thread communication and what is the class in which these methods are defined?

Multithreading – Technique where each set of code can be used by several processors at different stages of execution.

Methods for inter-thread communication:

public void wait() – waits until another thread invokes the notify()

public void notify() – wakes up single thread that’s waiting on objects monitor

public void notifyAll() – wakes up all threads that called wait() in same object

classes in which these methods are defined:

All these methods belong to an object class as final. They must be used within a synchronized block only.

1. What are Vector, Hashtable, LinkedList and Enumeration?

Vector: A data structure with the ability to implement a growing array of objects.

Hashtable: implements hashtable data structure. It uses hash codes as an object key to store objects in a dictionary. Hashtables are integer values that identify objects

LinkedList: to remove and add elements to the middle of an array, a linkedlinks would be the ideal data structure to use. A linkedlist stores each element in separate links whereas arrays will store the element as a while.

Enumeration: object implementing enumeration interface that generates a series of element one at a time. There are two methods that are used: hasMoreElements(), and nextElement(), HasMoreElemnts() tests.

1. What is a stream and what are the types of Streams and classes of the Streams?

Stream:

Types of streams:

Classes of the streams:

1. What is the difference between Reader/Writer and InputStream/Output Stream?-

Reader/Writer:

* Works at a character level where they read and write characters to the target class.

InputStream/Output Stream:

* Works at a byte level where they read and write bytes or list of bytes to a stream.

1. What is a Jar file?-

Jar (Java ARchive).

* Jar is a package file that is used for a file format based on the popular ZIP file format. Jar is used for aggregating multiple Java class files, associated metadata, and resources into one file to deliver it more efficiently as opposed to sending files one by one.

1. What is a Java Bean?-

Java Bean: Is a convention with lots of libraries that depend on it. It includes:

* Implement java.io.Seriializable interface – this is used to a save the state of an object
* Includes properties where its getters and setters are methods with certain names
* Uses an empty argument constructor to instantiate the object
* All JavaBean instance variables are all private