**CPS 450**

**HW 2 – Data Types**

**Date:Feb. 22, 2017**

**Total points: 30**

**Name\_\_\_\_*Mi Gao*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Qn. 1 (5 points)** Specify the ways in which user-defined enumeration types in C# are more reliable than those in C++.

C# enumeration types are like those of C++, except that they are never coerced to integer. So, operations on enumeration types are restricted to those that make sense. Also, the range of values is restricted to that of the particular enumeration type.

**Qn. 2 (5 Points)** Describe the lazy and eager approaches to reclaiming garbage (garbage collection), and list at least 1 programming language for each approach.

Reference counters (eager approach): reclamation is gradual.

Mark-sweep (lazy approach): reclamation occurs when the list of variable space becomes empty.

**Qn. 3 (5 Points)** What advantages do reference type variables as found in C# and Java have over the pointers in other languages like C and C++?

The reference Java and C# provide some of the flexibility and the capabilities of pointers, without the hazards.

**Qn. 4(5 points):** How does JavaScript support sparse arrays? What is a sparse array?

The value of subscripts need not to be contiguous.

A sparse array is one in which the elements do not have contiguous indexes starting at 0. Elements can be added at any index in an array so that gaps may result. Normally, the length property of an array specifies the number of elements in the array. If the array is sparse, the value of the length property is greater than the number of elements. Sparse arrays can be created with the Array() constructor or simply by assigning to an array index larger than the current array length.

**Qn. 5(5 Points):** Analyze and writer a comparison of C’s malloc and free functions with C++’s new and delete operators, using safety as a primary consideration in the comparison.

New and Delete are safe type. Malloc also return void which must then be thrown to the appropriate pointer type. New returns the correct pointer type by itself. Malloc requires you to tell the number of bytes that will be used to allocated. New will generate the numbers themselves.

**Qn. 6 (5 Points):** Compare the string manipulation capabilities of the class libraries of C++, Java, C#, and Python 3. Make note of how strings are represented, and of methods for doing comparison and for ignoring cases. Also make note of regular expression support.

C++: Not primitive; use char arrays and a library of functions that provide operations.

Python: Primitive type with assignment and several operations.

Java and C#: Primitive via the String class.

Example Methods:

C#:

string val = "AStringValue";

if (val.Equals("astringvalue", StringComparison.InvariantCultureIgnoreCase))

Java:

public class Test {

public static void main(String args[]) {

String Str1 = new String("This is really not immutable!!");

String Str2 = Str1;

String Str3 = new String("This is really not immutable!!");

String Str4 = new String("This IS REALLY NOT IMMUTABLE!!");

boolean retVal;

retVal = Str1.equals( Str2 );

System.out.println("Returned Value = " + retVal );

retVal = Str1.equals( Str3 );

System.out.println("Returned Value = " + retVal );

retVal = Str1.equalsIgnoreCase( Str4 );

System.out.println("Returned Value = " + retVal );

}

}

C++:

#include <boost/algorithm/string.hpp>

// Or, for fewer header dependencies:

//#include <boost/algorithm/string/predicate.hpp>

std::string str1 = "hello, world!";

std::string str2 = "HELLO, WORLD!";

if (boost::iequals(str1, str2))

{

// Strings are identical

}

Python: (Assuming ASCII strings)

string1 = 'Hello'

string2 = 'hello'

if string1.lower() == string2.lower():

print "The strings are the same (case insensitive)"

else:

print "The strings are not the same (case insensitive)"

Regular expression support:

The standard C++ library provides support for regular expressions in the <regex> header through a series of operations.

C# supports regular expressions through the classes in the System.Text.RegularExpressions namespace in the standard .NET framework.

Java based on java.util.regex. myString.matches("regex") returns true or false depending whether the string can be matched entirely by the regular expression.

The module **re** provides full support for Perl-like regular expressions in Python. The **re** module raises the exception **re.error** if an error occurs while compiling or using a regular expression.