# Final Exam

**due 14-Dec-2022 11:59pm**

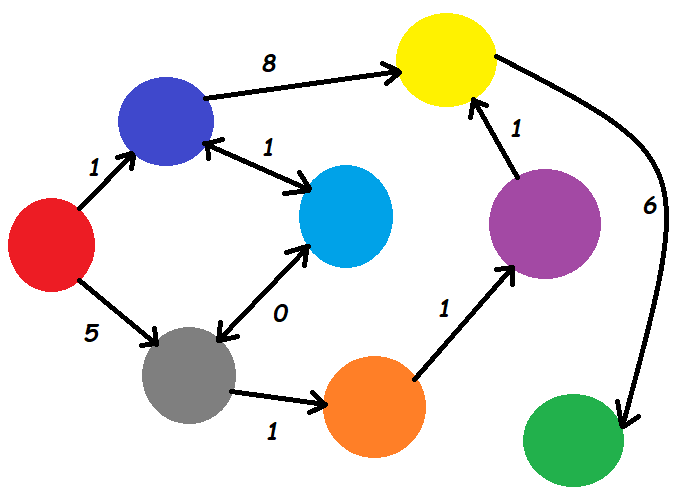
1. C# provides us with a Stack<int> generic template to create a stack of integers. Write your own MyStack class in a console app that contains a List<int> and implements the three Stack methods: Push(int n), Pop() and Peek() using the List<int>. You may not use Stack<int> in your solution.

GitHub URL:

1. Modify the console app in #1 to create your own MyQueue class that contains a List<int> and implements the three Queue methods: Enqueue(int n), Dequeue() and Peek() using the List<int>. You may not use Queue<int> in your solution.

GitHub URL:

1. Create a console application and represent the following digraph as an adjacency matrix and an adjacency list:



GitHub URL:

1. Using the console application created in #3, Implement a Depth First Search of this digraph starting from red and output the colors.

GitHub URL:

1. Using the console application created in #3, implement Dijkstra's shortest path algorithm to output the shortest path of colors from red to green.

GitHub URL:

1. Create a console application which defines and uses a singleton class which includes methods to load and save the player's settings using the following JSON format to a hard drive file, serializing and deserializing the structure with the Newtonsoft JSON package:

{"player\_name":"dschuh","level":4,"hp":99,"inventory":["spear","water bottle","hammer","sonic screwdriver","cannonball","wood","Scooby snack","Hydra","poisonous potato","dead bush","repair powder"],"license\_key":"DFGU99-1454"}

GitHub URL:

1. Which 2 specific traversal methods can be used to make an exact copy of a binary search tree?

Pre-Order and Post-Order.

1. True or **False**: This code will copy the first node to the end of a LinkedList

LinkedListNode<string> firstNode = sentence.First;

sentence.AddLast(firstNode);

If false, how would you write it correctly?

This is incorrect because it would pass the reference of the node instead of making a new copy. The correct code could be:

LinkedListNode<string> firstNode = new LinkedListNode<string>(sentence.First);

sentence.AddLast(firstNode);

1. Complete the code below to have the myRounder delegate method point to Math.Round(double d, int n). Given all the abbreviated lambda expressions, there are a total of 12 ways to write the code. An extra point for each additional solution (up to 11 extra points are available).

namespace DelegateFunction

{

class Program

{

public delegate double MyRounder(double d, int n);

static void Main(string[] args)

{

// create variable of delegate function type

MyRounder myRounder;

// Solution 1

myRounder = new MyRounder(Math.Round);

// Solution 2  
myRounder = Math.Round;

// Solution 3  
myRounder = delegate (double d, int n)  
{  
double returnVal = Math.Round(d, n);  
return returnVal;  
};

// Solution 4  
myRounder = (double d, int n) =>  
{  
double returnVal = Math.Round(d, n);  
return returnVal;  
};

// Solution 5  
myRounder = (d, n) =>  
{  
double returnVal = Math.Round(d, n);  
return returnVal;  
};

// Solution 6  
myRounder = (d, n) => Math.Round(d, n);  
Func<double, int, double> myGenericRounder;

// Solution 7  
myGenericRounder = new Func<double, int, double>(Math.Round);

// Solution 8  
myGenericRounder = Math.Round;

// Solution 9  
myGenericRounder = delegate (double d, int n)  
{  
double returnVal = Math.Round(d, n);  
return returnVal;  
};

// Solution 10  
myGenericRounder = (double d, int n) =>  
{  
double returnVal = Math.Round(d, n);  
return returnVal;  
};

// Solution 11  
myGenericRounder = (d, n) =>  
{  
double returnVal = Math.Round(d, n);  
return returnVal;  
};

// Solution 12  
myGenericRounder = (d, n) => Math.Round(d, n);

}

}

}

1. True or false:
   1. an abstract class can be instantiated **False**
   2. a sealed class can be inherited **False**
   3. a property can have a get function or a set function, but not both **False**
   4. an interface can only include methods and properties **True**
   5. a class can only directly inherit one class **True**
   6. private fields in a parent class are inherited by a child class **False**
   7. protected members of a parent class are inherited by a child class **True**
   8. a static class cannot be instantiated **True**
   9. a class constructor can be private **True**
   10. struct is a "by value" data type **True**
   11. class is a "by value" data type **False**

## Submission

Upload this completed document to the corresponding myCourses dropbox.