**CS345 Sample Test**

1. Consider the following function:

void someFunction(ref int first, out float second, ref string third, SomeClass fourth,

 int fifth, SomeStruct sixth)

{

 first = 11;

 second = 12;

 third = “13”;

 fourth .MyProp = 14;

 fifth = 15;

 sixth.myInt\_ = 16;

}

 class SomeClass

 {

 public int MyProp {get; set;}

 }

 struct SomeStruct

 {

 public int myInt\_;

 }

 int a = 1;

 float b = 2.0f;

 string c = “c”;

 SomeClass d = new SomeClass();

 d.MyProp = 4;

 int e = 5;

 SomeStruct f;

 f.myInt = 6;

* + Write the call to send variables “a” through “f” to “first” through “sixth” respectively:

Answer:

someFunction(ref a, out b, ref c, d, e, f);

* + What are the values of the variables upon return?

a is 11

b is 12

c is “13”

d .MyProp is 14

e is 5

f.myInt\_ is 6

1. Write C# to allocate an array of ints called myArray containing the numbers 1 through 12

Answer:

int[] myArray = new int[] {1,2,3,4,5,6,7,8,9,10,11,12};

1. Write a foreach loop to print each of the items in myArray

Answer:

foreach(int i in myArray)

{

 Console.WriteLine(“{0}”, i);
}

1. Declare a delegate called someDelegate. Make an instance of that delegate called myDelegate and point to the function int max(int a, int b). Invoke max(10,20) using the delegate.

 static int max(int a, int b)

 {

 return (a > b) ? a : b;

 }

 delegate int someDelegate(int a, int b);

 someDelegate myDelegate = max;

 int y = myDelegate(10,20);

1. Declare an event called someEvent uses the delegate in question 4.
	1. Have it to point to the max and
	2. use it to find the max of 100 and 200.

// Declare the event as a data member of a class

event someDelegate someEvent;

 // In one of the class members, add max as an event handler

 someEvent += max;

 // Cause the event handler to fire.

 int y = someEvent(100,200);

1. Name some events that are typically handled by a Windows application

Answer:

* + MouseClick
	+ MouseMove
	+ Paint
	+ Resize
1. What are reference types? How are they different from value types? Give some examples of each.

Answer:

Reference types are implemented as handles to dynamically allocated (garbage collected) memory. Value types are implemented much like C++ variables (i.e., without indirection) on the run-time stack.

All classes are reference types in C#. string is another example of a reference type.

All structures are value types in C#. int, float, and most of the primitive data types are value types.

1. What is the purpose of the Invalidate() function?

Answer: It generates a Paint event to cause the screen to be redrawn.

1. What are resources and why should we use them?

Answer: They are strings, icons, images, and other data that, rather than being stored in files, are embedded into the assembly that comprises the executable program. They simplify many tasks including installation (external files do not need to accompany the executable file, and internationalization (because text the user sees is not hard coded, and ports to different languages can be accomplished through changes to the resources).

1. What are world coordinates?

They are the coordinates that correspond to the units used in application domain. For example, if we are drawing a football field, world coordinates would probably be expressed in yards. If we are displaying a football field in a window that is 600 pixels wide and 300 pixels high, the device coordinates would range from (0,0) to (600,300), pixels but the world coordinates might range from (0,0) to (100,50) yards.

1. What are the advantages of using control containers such as ToolStripContainer and SplitContainer?

Answer: They free us from the chores of determining the exact locations of our components and provide the opportunity to allow controls to be automatically or semi-automatically placed.

1. Describe the DockStyles Left, Right, and Fill.

DockStyle.Left and DockStyle.Right cause the control to get as close as it can to the Left and Right sides respectively. DockStyle.Fill uses all the available space.

1. What is padding?

Padding is the internal distance in pixels from the edge of a forms or control’s client area that will not be used for automatically placed controls. Padding can be specified separately for each of the sides (Bottom, Top, Left, Right).

1. What is anchoring?

 Anchoring is a technique used to preserve the distance between a control and the edge of its container as the container is resized. A control can be anchored to the Top, Bottom, Left, and/or Right of the container. If a control is anchored to two opposing sides, it will grow as the container is resized.

1. Create an abstract base class called MyBase with an abstract function called someFunction() and a virtual function called anotherFunction(). Write a subclass called MySub that overrides both functions. Write a constructor for the MyBase class that takes a string name as a parameter. Write a constructor for the MySub class that takes a string name as a parameter and use that string to invoke the MyBase class’s constructor.

abstract class MyBase

 {

 public MyBase(string name) { name\_ = name; }

 public abstract void someFunction();

 public virtual void anotherFunction() {}

 private string name\_;

 }

 class MySub : MyBase

 {

 public MySub(string name) : base(name)

 {}

 public override void someFunction()

 {

 }

 public override void anotherFunction()

 {

 }

 }