1. What is Encapsulation?

Is wrapping variables and methods into a single unit.

1. What is Polymorphism?

Is redefining of method for derived classes through inheritance.

1. What is Inheritance?

Taking properties existing objects.

1. What is Abstraction?

Is hidden the implementation of an object and only exposing object definition of exposed properties and methods through an interface.

1. Why do we use Classes?

Are used to define object’s data structure that hold inform about something and are key to OOP

1. What is an Interface? (NOT UI!)

Defines the interaction between components without understanding the implementation of the provided methods.

1. What are Virtual Methods and why are they useful

Virtual methods are methods that can be defined by subclass, they are helpful by providing ability to be overridden by a subclass implementing the same method.

1. What are the SOLID principles?

S : Single responsibility principle

O : Open closed principle

L : Liskov substitution principle

I : Interface segregation principle

D : Dependency inversion principle

1. Why is Source Control important?

Maintain source code in a secure environment, manages source code versions and maintains history of the source code.

1. What happens when the following is executed?

int a = 10;

int b = 0;

int c = a/b;

Console.WriteLine(“The answer is “+ c.ToString());

It will throw an exception “System.DivideByZeroException”

1. What are Unit Tests and why are they essential?

The development process to small testable parts of the application. It helps with detecting the changes on you code which may results in breaking other parts of the software and reduces bugs.

1. What are try, catch, finally used for?

Exception handling

1. What are Generics? Why are they useful?

Is have a class/interface with placeholder to define a type at runtime.

Improves the performance on collection of objects

1. What are DataContract, DataMember, OperationContract used for (hint: Distributed Services):

They are WCF service attributes.  
DataContract agrees on the data to be exchanged between server and client.

DataMember determines which properties are part for datatcontract and can be accessed on the WCF

OperationContract determines which methods are exposed to a service.

1. Will the following method be accessible from outside an ASMX WebService?

public int GetAmountOfDaysWorked(string workerIdNo){…}

No, you must specify **WebMethod** attribute

1. Give two (2) different ways the keyword “using” can be used:

using System.Text;

using(var obj = new object()){

}

1. In the context of software development, what is recursion?

Is have a same function/ method call itself and must have base case to stop.

1. Difference between while(…){…}, do{…} while(…)?

while first checks the test expression if is true first and do while execute loop code then checks then checks the expression last.

1. Given the following tables, write the SQL statement(s) that returns Qualifying Customer and their Items.

**Customers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID(int) | Customer(varchar(50)) | Email(varchar(50)) | Age(int) | Qualifies(bit) |
| 1 | Matthew | [matthew@abc.com](mailto:matthew@abc.com) | 18 | 0 |
| 2 | Mark | [mark@abc.com](mailto:mark@abc.com) | 21 | 1 |
| 3 | Luke | [luke@abc.com](mailto:luke@abc.com) | 23 | 1 |
| 4 | John | [john@abc.com](mailto:john@abc.com) | 25 | 0 |
| 5 | Paul | [paul@abc.com](mailto:paul@abc.com) | 29 | 1 |

**Orders**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID(int) | CustomerID(int) | Item(varchar(max)) | Price(decimal(18,2)) | Date(DateTime) |
| 1 | 1 | Coke | 16.00 | 2018-01-01 |
| 2 | 2 | Pepsi | 14.00 | 2018-01-02 |
| 3 | 3 | 7-Up | 13.00 | 2018-01-03 |
| 4 | 4 | Sprite | 14.00 | 2018-01-04 |
| 5 | 5 | Fanta | 16.00 | 2018-01-05 |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Qualifying customers**

SELECT ID,Customer, Email,Age FROM Customers WHERE Qualifies=1

**Qualifying customers orders**

Orders

SELECT [order].ID,Item, Price,[Date] FROM Customers customer

INNER JOIN Orders [order] ON [order].ID = customer.ID

WHERE Qualifies=1

1. Select all the names containing “a”, order them in alphabetical order using LINQ:

string[] names = new string[]

{“Tom”, “Paul”, “Sally”, “Chris”, “Peter”, “Bruce”, “Clark”};

names.Where(x=>x.Contains("a")).OrderBy(y=>y);