1) WHAT IS THE DIFFERENCE BETWEEN COMPILATION AND DEBUGGING?

Debugging: -Laptop programmers, like all people else, will not be excellent. This implies the applications they write generally have small errors, referred to as "bugs," in them. These bugs could be minor, resembling not recognizing person enter, or extra critical, resembling a reminiscence leak that crashes this system. Earlier than releasing their software program to the general public, programmers "debug" their applications, eliminating as many errors as attainable. This debugging course of typically takes a very long time, as fixing some errors could introduce others.

Compiling: -When programmers create software program applications, they first write this system in supply code, which is written in a particular programming language, resembling C or Java. These supply code records data are saved in a text-based, human-readable format, which could be opened and edited by programmers. Nevertheless, the supply code can't be run straight by the pc. To ensure that the code to be understood by the pc's CPU, it has to be compiled into an executable program.

2) WHAT IS THE DIFFERENCE BETWEEN ELSE IF LADDER AND SWITCH CASE?

- In else if ladder, the control goes through the every else if statement until it finds true value of the statement or it comes to the end of the else if ladder. In case of switch case, as per the value of the switch, the control jumps to the corresponding case.
- The switch case is more compact than lot of nested else if. So, switch is considered to be more readable.
- The use of break statement in switch is essential but there is no need of use of break in else if ladder.
- The variable data type that can be used in expression of switch is integer only where as in else if ladder accepts integer type as well as character.
- Another difference between switch case and else if ladder is that the switch statement is considered to be less flexible than the else if ladder, because it allows only testing of a single expression against a list of discrete values.
- Since the compiler is capable of optimizing the switch statement, they are generally considered to be more efficient. Each case in switch statement is independent of the previous one. In case of else if ladder, the code needs to be processed in the order determined by the programmer.
- Switch case statement work on the basis of equality operator whereas else if ladder works on the basis of true false(zero/non-zero) basis.

3) WHAT IS DIFFERENCE AMONG ALL TYPES OF ITERATIVE STATEMENT?

While Statement: It is a simple loop. It executes till the code block while the condition is true. If the condition is going to be false, the program will exit from the loop.

Do-While Statement: The do while loop is same as while loop except that it will be running at least one time if condition is matched or not. It is because it does not check the condition first time. So, it is guaranteed to execute the code of program at least one iteration.

For Statement: The For loop is used if you know the start point and end point. You can run a statement or a block of statements repeatedly until a specified expression evaluates too false. It is useful where you know in advance how many times program should iterate.

For each Statement: The foreach statement repeats a group of embedded statements for each element in an array or an object collection. It means if you are working on the array or collection of objects than you should use foreach statement to iterate. So, foreach statement iterates every element of collection and forwards it to next statement.

4) WHAT IS THE DIFFERENCE BETWEEN (a++ POSTFIX, ++a PREFIX)?

a++ is known as postfix.

add 1 to a, returns the old value.

++a is known as prefix.

add 1 to a, returns the new value.

```
string [] items = {"a","b","c","d"};
int i = 0;
foreach (string item in items)
{
Console.WriteLine(++i);
}
Console.WriteLine("");
i = 0;
foreach (string item in items)
{
Console.WriteLine(i++);
}
```

Output:

1

2

3

4

0

1

2

3

5) DIFFERENCE BETWEEN ABSTRACT CLASS, SEALED CLASS, INTERFACE?

Abstract class: Should be used when there is an IS-A relationship and no instances should be allowed to be created from that abstract class. Example: An Animal is an abstract base class where specific animals can be derived from, i.e., Horse, Pig etc. By making Animal abstract it is not allowed to create an Animal instance.

Interface: An interface should be used to implement functionality in a class. Suppose we want a horse to be able to Jump, an interface I Jumping can be created. By adding this interface to Horse, all methods in IJumping should be implemented. In IJumping itself only the declarations (e.g. StartJump and EndJump are defined), in Horse the implementations of these two methods should be added.

Sealed class: By making Horse sealed, it is not possible to inherit from it, e.g., making classes like Pony or Workhorse which you like to be inheriting from Horse.

6) WHAT IS NESTED CLASS WITH ONE EXAMPLE?

A class is a user-defined blueprint or prototype from which objects are created. Basically, a class combines the fields and methods (member function which defines actions) into a single unit. In C#, a user is allowed to define a class within another class. Such types of classes are known as nested class. This feature enables the user to logically group classes that are only used in one place, thus this increases the use of encapsulation, and create more readable and maintainable code.

Syntax:

```
class Outer_class {
    // Code..
    class Inner_class {
      // Code..
    }
}
```

7) WHAT IS XML? WHY XML,WHAT IS XSD,DTD,XPATH and Schema?

What is XML

- XML stands for extensible Markup Language.
- XML was designed to store and transport data.
- XML was designed to be both human- and machine-readable.

WHY XML

- XML plays an important role in many different IT systems.
- XML is often used for distributing data over the Internet.
- It is important (for all types of software developers!) to have a good understanding of XML.

What is XPath?

- XPath is a major element in the XSLT standard.
- XPath can be used to navigate through elements and attributes in an XML document. Path is a syntax for defining parts of an XML document
- XPath uses path expressions to navigate in XML documents
- XPath contains a library of standard functions
- XPath is a major element in XSLT and in XQuery

What is a DTD?

- DTD stands for Document Type Definition.
- A DTD defines the structure and the legal elements and attributes of an XML document.

What is Schema

- An XML Schema describes the structure of an XML document, just like a DTD.
- An XML document with correct syntax is called "Well Formed".
- An XML document validated against an XML Schema is both "Well Formed" and "Valid".

8) Difference between Document Type Definition (DTD) and XML Schema Definition (XSD) :

Sr No	DTD	XSD
1	DTD are the declarations that define a document type for SGML.	XSD describes the elements in a XML document.
2	It doesn't support namespace.	It supports namespace.
3	It is comparatively harder than XSD.	It is relatively simpler than DTD.
4	It doesn't support datatypes.	It supports datatypes.
5	SGML syntax is used for DTD.	XML is used for writing XSD.
6	It is not extensible in nature.	It is extensible in nature.
7	It doesn't give us much control on structure of XML document.	It gives us more control on structure of XML document.

9) PCDATA AND CDATA IN DTD?

CDATA: (Unparsed Character data): CDATA contains the text which is not parsed further in an XML document. Tags inside the CDATA text are not treated as markup and entities will not be expanded.

PCDATA: (Parsed Character Data): XML parsers are used to parse all the text in an XML document. PCDATA stands for Parsed Character data. PCDATA is the text that will be parsed by a parser. Tags inside the PCDATA will be treated as markup and entities will be expanded.

10) ONE LINE DEFINITION FOR ALL TYPE EXCEPTION?

Argument Exception :->

Raised when a non-null argument that is passed to a method is invalid.

ArgumentNullException :->

Raised when null argument is passed to a method.

ArgumentOutOfRangeException :->

Raised when the value of an argument is outside the range of valid values.

DivideByZeroException:->

Raised when an integer value is divided by zero.

FileNotFoundException:->

Raised when a physical file does not exist at the specified location.

FormatException:->

Raised when a value is not in an appropriate format to be converted from a string by a conversion method such as Parse.

IndexOutOfRangeException:->

Raised when an array index is outside the lower or upper bounds of an array or collection.

InvalidOperationException:->

Raised when a method call is invalid in an object's current state.

KeyNotFoundException:->

Raised when the specified key for accessing a member in a collection is not exists.

NotSupportedException :->

Raised when a method or operation is not supported.

NullReferenceException :->

Raised when program access members of null object.

OverflowException:->

Raised when an arithmetic, casting, or conversion operation results in an overflow.

OutOfMemoryException:->

Raised when a program does not get enough memory to execute the code.

StackOverflowException :->

Raised when a stack in memory overflows.

TimeoutException:->

The time interval allotted to an operation has expired.

11) DIFFERENCE BETWEEN THROW AND THROWS?

Throw: If we use "throw" statement, it preserves original error stack information. In exception handling "throw" with empty parameter is also called re-throwing the last exception.

Throw ex: If we use "throw ex" statement, stack trace of exception will be replaced with a stack trace starting at the re-throw point. It is used to intentionally hide stack trace information.

12) FIND OUT DIFFERENCE BETWEEN THREADSTART AND THREAD POOLWAY FOR IMPLEMENTATION FOR MULTITHREADING?

Thread Pool:

Thread pool in C# is nothing but a collection of threads that can be reused to perform no of tasks in the background. Now when a request comes, then it directly goes to the thread pool and checks whether there are any free threads available or not. If available, then it takes the thread object from the thread pool and executes the task.

```
using System;

using System.Diagnostics;

using System.Threading;

namespace ThreadPoolApplication

{

   class Program

   {

      static void Main(string[] args)

      {

         for (int i = 0; i < 10; i++)
```

```
    {
        MethodWithThread();
        MethodWithThreadPool();
    }
    Stopwatch stopwatch = new Stopwatch();
    Console.WriteLine("Execution using Thread");
    stopwatch.Start();
    MethodWithThread();
    stopwatch.Stop();
    Console.WriteLine("Time consumed by MethodWithThread is : " +
                      stopwatch.ElapsedTicks.ToString());

    stopwatch.Reset();
    Console.WriteLine("Execution using Thread Pool");
    stopwatch.Start();
    MethodWithThreadPool();
    stopwatch.Stop();
    Console.WriteLine("Time consumed by MethodWithThreadPool is : " +
                      stopwatch.ElapsedTicks.ToString());

    Console.Read();
}
public static void MethodWithThread()
{
    for (int i = 0; i < 10; i++)
    {
        Thread thread = new Thread(Test);
    }
}
public static void MethodWithThreadPool()
```

```
    {
        for (int i = 0; i < 10; i++)
        {
            ThreadPool.QueueUserWorkItem(new WaitCallback(Test));
        }
    }
    public static void Test(object obj)
    {
    }
  }
}
```

THREAD START:

When a program calls the start() method, a new thread is created and then the run() method is executed. But if we directly call the run() method then no new thread will be created and run() method will be executed as a normal method call on the current calling thread itself and no multi-threading will take place.

Let us understand it with an example:

```
class MyThread extends Thread {
  public void run()
  {
    System.out.println("Current thread name: "
                + Thread.currentThread().getName());
    System.out.println("run() method called");
  }
}

class G1 {
  public static void main(String[] args)
  {
    MyThread t = new MyThread();
```

```
        t.start();
    }
}
```

13) DIFFERENCE BETWEEN MULTITHREADING AND PARALLEL PROGRAMMING?

Multithreading: On a single processor, multithreading gives the illusion of running in parallel. In reality, the processor is switching by using a scheduling algorithm. Or, it's switching based on a combination of external inputs (interrupts) and how the threads have been prioritized.

Parallel programming: Parallel programming is a programming technique wherein the execution flow of the application is broken up into pieces that will be done at the same time (concurrently) by multiple cores, processors, or computers for the sake of better performance.

14) DIFFERENCE BETWEEN DESTRUCTOR, FINALIZE AND DISPOSE?

Finalize:

Finalize () is called by the Garbage Collector before an object that is eligible for collection is reclaimed. Garbage collector will take the responsibility to deallocate the memory for the unreferenced object. The Garbage Collector calls this method at some point after there are no longer valid references to that object in memory.

Dispose:

There are some resources like windows handles, database connections, network connections, files, etc. which cannot be collected by the Garbage Collector. If we want to explicitly release some specific objects then this is the best to implement IDisposable and override the Dispose() method of Disposable interface. The Dispose() method is not called automatically and we must explicitly call it from a client application when an object is no longer needed. Dispose() can be called even if other references to the object are alive.

Destructor:

destructors can be used only in classes, and a class can contain only one destructor. The destructor in class can be represented by using the tilde (~) operator The destructor in c# won't accept any parameters and access modifiers. The destructor will invoke automatically whenever an instance of a class is no longer needed. The destructor is automatically invoked by the garbage collector whenever the class objects are no longer needed in the application.

15) WRITE DIFFERENCE BETWEEN TRUNCATE AND DELETE?

Sr No	Delete	Truncate
1	The DELETE command is used to delete specified rows(one or more).	While this command is used to delete all the rows from a table.
2	It is a DML(Data Manipulation Language) command.	While it is a DDL(Data Definition Language) command.
3	There may be WHERE clause in DELETE command in order to filter the records.	While there may not be WHERE clause in TRUNCATE command.
4	In the DELETE command, a tuple is locked before removing it.	While in this command, data page is locked before removing the table data.

16) WRITE DIFFERENCE BETWEEN TRUNCATE AND DROP?

Sr No	DROP	TRUNCATE
1	The DROP command is used to remove table definition and its contents.	Whereas the TRUNCATE command is used to delete all the rows from the table.
2	In the DROP command, table space is freed from memory.	While the TRUNCATE command does not free the table space from memory.
3	DROP is a DDL(Data Definition Language) command.	Whereas the TRUNCATE is also a DDL(Data Definition Language) command.
4	In the DROP command, view of table does not exist.	While in this command, view of table exist.
5	In the DROP command, integrity constraints will be removed.	While in this command, integrity constraints will not be removed.
6	In the DROP command, undo space is not used.	While in this command, undo space is used but less than DELETE.
7	The DROP command is quick to perform but gives rise to complications	While this command is faster than DROP.

17) WRITE DOWN THE AMONG TRUNCATE,DELETE,DROP?

- DELETE:

  Basically, it is a Data Manipulation Language Command (DML). It is use to delete the one or more tuples of a table. With the help of "DELETE" command we can either delete all the rows in one go or can delete row one by one. i.e., we can use it as per the requirement or the condition using Where clause. It is comparatively slower than TRUNCATE cmd.

SYNTAX –

If we want to delete all the rows of the table:

DELETE from;

- DROP :

  It is a Data Definition Language Command (DDL). It is use to drop the whole table. With the help of "DROP" command we can drop (delete) the whole structure in one go i.e., it removes the named elements of the schema. By using this command, the existence of the whole table is finished or say lost.

SYNTAX –

If we want to drop the table:

DROP table ;

- . TRUNCATE:

  It is also a Data Definition Language Command (DDL). It is use to delete all the rows of a relation (table) in one go. With the help of "TRUNCATE" command we can't delete the single row as here WHERE clause is not used. By using this command the existence of all the rows of the table is lost. It is comparatively faster than delete command as it deletes all the rows fastly.

SYNTAX –

If we want to use truncate :

TRUNCATE ;

18) FIND OUT THE RANGE OF DATA TYPES?

int - -2,147,483,648, to 2,147,483,647

char - 8000 characters

varchar - 8000 characters

varchar(max) - 1,073,741,824

text - 2GB of textdata

nchar - 4000 charcters

nchar - 4000 characters

19) ONE LINE DEFINITION ENTERPRISE STANDARD DEVELOPER EXPRESS?

Enterprise Evaluation Edition:

Enterprise Evaluation Edition is a free (for 180 days) edition. All features are the same Product as Enterprise Edition.

Express Edition:

Express Edition is a scaled-down edition of SQL with some limitations but allowed for production use.

Developer Edition:

Developer Edition is a free edition (starting SQL 2016 onwards) which is allowed only for development but not for production purposes. All features are the same as Enterprise Edition.

SQL Server:

SQL Server Standard edition delivers basic data management and business intelligence database for departments and small organizations to run their applications and supports common development tools for on-premises and cloud - enabling effective database management with minimal IT resources.