1)What is the difference between compilation and debugging?

Compiler :

Compiler, as name suggests, is a process that is used to convert code into machine instructions. It simply translates source code from high-level programming language to low-level machine language. It is basically a complex software that performs both code optimization and code generation. It also makes end code more efficient which is optimized for execution time and memory space.

Debugger :

Debugger, as name suggests, is a process used to remove bugs from code. It simply allows testing and debugging other programs. Sometime, it also provides two modes of operations i.e., full and partial simulation. It is used to prevent incorrect operation of software or system. It also uses instruction-set simulators instead of running a program directly on processor to achieve higher level of control over its execution.

2)What is the difference between else if ladder and switch case?

Else if ladder :

In else if ladder, the control runs through the every else if statement until it arrives at the true value of the statement or until it comes to the end of the else if ladder.

Else if ladder statement works on the basis of true false (zero/non-zero) basis.

In else if ladder, the use of break statement is not very essential

In the case of else if ladder, the code needs to be processed in the order determined by the programmer.

Switch case :

In else if ladder, the control runs through the every else if statement until it arrives at the true value of the statement or until it comes to the end of the else if ladder.

Switch case statement work on the basis of equality operator.

In switch, the use of break statement is mandatory and very important.

In switch case, it is possible to optimize the switch statement, because of their efficiency. Each case in switch statement does not depend on the previous one.

3)What is the difference among all types of iterative statements?

while statement

The while loop in C is most fundamental loop statement. It repeats a statement or block while its controlling expression is true. Here is its general form:

while(condition) { // body of loop }

The condition can be any boolean expression. The body of the loop will be executed as long as the conditional expression is true.

Here is more practical example.

```
//example program to illustrate while looping
 #include <stdio.h>
 int main ()
{
 int n = 10;
while (n > 0)
{
printf("tick %d", n);
printf("\n");
n--;    }
}
```

The ouput of the program is:

tick 10 tick 9 tick 8 tick 7 tick 6 tick 5 tick 4 tick 3 tick 2 tick 1

2. **do-while statement**

The do-while loop always executes its body at least once, because its conditional expression is at the bottom of the loop. Its general form is:

do{ // body of loop } while(condition);

Each iteration of the do-while loop first executes the body of the loop and then evaluates the conditional expression. If this expression is true, the loop will

repeat. Otherwise, the loop terminates. As with all of C's loops, condition must be a Boolean expression.

The program presented in previous while statement can be re-written using do-while as:

```
//example program to illustrate do-while looping
#include <stdio.h>
int main ()
{
int n = 10;
do
{
printf("tick %d", n);
printf("\n");
n--;
}while (n > 0);
}
```

3. for loop

Here is the general form of the traditional for statement:

for(initialization; condition; iteration) { // body }

The program from previous example can be re-written using for loop as:

```
//example program to illustrate for looping
 #include <stdio.h>
int main ()
{
int n;
for(n = 10; n>0 ; n--)
{
printf("tick %d",n);
printf("\n");
```

```
}
}
```

The ouput of the program is same as output from program in while loop.

There can be more than one statement in initilaization and iteration section. They must be separated with comma.

Here, we've presented the example to illustrate this.

```
//example program to illustrate more than one statement using the comma // for looping
#include <stdio.h>
int main ()
{
int a, b;
for (a = 1, b = 4; a < b; a++, b--)
{
printf("a = %d \n", a);
printf("b = %d \n", b);
}
}
```

The ouput of the program is:

```
a = 1 b = 4 a = 2 b = 3
```

4)What is the difference (a++ (Postfix),++a (Prefix))?

The postfix increment ++ does not increase the value of its operand until after it has been evaluated. The value of i++ is i.

The prefix decrement increases the value of its operand before it has been evaluated. The value of --i is i - 1.

Prefix increment/decrement change the value before the expression is evaluated. Postfix increment/decrement change the value after.

5)What is the difference between for and foreach loop?

For Loop :

The for loop executes a statement or a block of statements repeatedly until a specified expression evaluates to false.

There is need to specify the loop bounds (minimum or maximum). Following is a code example of a simple for loop that starts 0 till <= 5.

```
int j = 0;

for (int i = 1; i <= 5; i++)

{

  j = j + i ;

}
```

Foreach Loop :

The foreach statement repeats a group of embedded statements for each element in an array or an object collection. You do not need to specify the loop bounds minimum or maximum. The following code loops through all items of an array.

```
int j = 0;

int[] myArr = new int[] { 0, 1, 2, 3, 5, 8, 13 };

foreach (int i in myArr )

{

  j = j + i ;

}
```

6)Find the difference between interface, abstract class and sealed class?

abstract class

Should be used when there is a 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 IJumping 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.

7)Difference between DTD and XSD?

No.	DTD	XSD
1)	DTD stands for Document Type Definition.	XSD stands for XML Schema Definition.
2)	DTDs are derived from SGML syntax.	XSDs are written in XML.
3)	DTD doesn't support datatypes.	XSD supports datatypes for elements and attributes.
4)	DTD doesn't support namespace.	XSD supports namespace.
5)	DTD doesn't define order for child elements.	XSD defines order for child elements.
6)	DTD is not extensible.	XSD is extensible.
7)	DTD is not simple to learn.	XSD is simple to learn because you don't need to learn language.
8)	DTD provides less control on XML structure.	XSD provides more control on XML structure.

8)PCDATA and CDATA in DTD

PCDATA is text that will be parsed by a parser. Tags inside the text will be treated as markup and entities will be expanded.

CDATA is text that will not be parsed by a parser. Tags inside the text will *not* be treated as markup and entities will not be expanded.

9)Difference between Throw and Throws with implementation in program.?

Throw:

keyword is used inside a function. It is used when it is required to throw an Exception logically.

```
void Demo() throws ArithmeticException, NullPointerException

{

// Statements where exceptions might occur.

throw new ArithmeticException();

}
```

Throws:

keyword is in the function signature. It is used when the function has some statements that can lead to some exceptions.

```
void Demo(){

// Statements where exceptions might occur.}
```

10) One line definition for different types of exceptions?

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 value
DivideByZeroException	: Raised when an integer value is divide 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 fro string by a conversion method such as Parse.
IndexOutOfRangeException	: Raised when an array index is outside the lower or upper bounds of an a 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 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 i overflow.
OutOfMemoryException	: Raised when a program does not get enough memory to execute the co
StackOverflowException	: Raised when a stack in memory overflows.
TimeoutException	: The time interval allotted to an operation has expired.

11)Find out the difference between Thread start and thread pool way of implementation for multithreading.

ThreadPool:
Many applications create threads that spend a great deal of time in the sleeping state, waiting for an event to occur. Other threads might enter a sleeping state only to be awakened periodically to poll for a change or update status information. The thread pool enables you to use threads more efficiently by providing your application with a pool of worker threads that are managed by the system.

```
using System;

using System.Threading;



public class Example

{

   public static void Main()

   {

      // Queue the task.

      ThreadPool.QueueUserWorkItem(ThreadProc);
```

```
        Console.WriteLine("Main thread does some work, then sleeps.");
        Thread.Sleep(1000);

        Console.WriteLine("Main thread exits.");
    }

    static void ThreadProc(Object stateInfo)
    {
            Console.WriteLine("Hello from the thread pool.");
    }
```

ThreadStart:

When a managed thread is created, the method that executes on the thread is represented by a ThreadStart delegate or a ParameterizedThreadStart delegate that is passed to the Thread constructor. The thread does not begin executing until the Thread.Start method is called. Execution begins at the first line of the method represented by the ThreadStart or ParameterizedThreadStart delegate.

```
using System;
using System.Threading;

class Test
{
    static void Main()
    {

        ThreadStart threadDelegate = new ThreadStart(Work.DoWork);
```

```
        Thread newThread = new Thread(threadDelegate);
        newThread.Start();

        Work w = new Work();
        w.Data = 42;
        threadDelegate = new ThreadStart(w.DoMoreWork);
        newThread = new Thread(threadDelegate);
        newThread.Start();
    }
}
class Work
{
    public static void DoWork()
    {
        Console.WriteLine("Static thread procedure.");
    }
    public int Data;
    public void DoMoreWork()
    {
 Console.WriteLine("Instance thread procedure. Data={0}", Data);
    }
}
```

12)Find out the difference between Destructor, finalize and Dispose

Destructor

They are special methods that contains clean up code for the object. You can't call them explicitly in your code as they are called implicitly by GC (Garbage Collector). In C# they have same name as the class name preceded by the "~" sign. Like~

Dispose
These are just like any other methods in the class and can be called explicitly but they have a special purpose of cleaning up the object. In the dispose method we write clean up code for the object. It is important that we freed up all the unmanaged resources in the dispose method like database connection, files etc.

Finalize
Finalize () is called by Garbage Collector implicitly to free unmanaged resources. The garbage collector calls this method at some point after there are no longer valid references to the object. There are some resources like windows handles, database connections which cannot be collected by the garbage collector. Therefore the programmer needs to call Dispose() method of IDisposable interface.
· Implement it when you have unmanaged resources in your code, and want to make sure that these resources are freed when the Garbage collection happens.

13)Difference between Multithreading and parallel programming
Multithreading: an environment/hardware architecture that allows the efficient execution of multiple threads.

Parallel programming: multiple threads solve simultaneously a problem to improve performances (e.g., searching the solution of a math problem, sort a list with quicksort)

14) Write difference between Truncate and delete

SQL Delete	**SQL Truncate**
Delete command is useful to delete all or specific rows from a table specified using a Where clause	The truncate command removes all rows of a table. We cannot use a Where clause in this.
It is a DML command	It is a DDL command.
SQL Delete command places lock on each row requires to delete from a table.	SQL Truncate command places a table and page lock to remove all records.

Delete command logs entry for each deleted row in the transaction log.	The truncate command does not log entries for each deleted row in the transaction log.
Delete command is slower than the Truncate command.	It is faster than the delete command.

15) Write difference between Truncate and Drop

SQL Drop	**SQL Truncate**
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.
In the DROP command, table space is freed from memory.	While the TRUNCATE command does not free the table space from memory.
DROP is a DDL(Data Definition Language) command.	Whereas the TRUNCATE is also a DDL(Data Definition Language) command.
In the DROP command, view of table does not exist.	While in this command, view of table exist.
In the DROP command, integrity constraints will be removed.	While in this command, integrity constraints will not be removed.
In the DROP command, undo space is not used.	While in this command, undo space is used but less than DELETE.
The DROP command is quick to perform but gives rise to complications.	While this command is faster than DROP.

16) Write down the among truncate, Delete and drop

Truncate:

Truncate Command is a Data Definition Language operation. It is used to remove all the records from a table. It deletes all the records from an existing table but not the table itself. The structure or schema of the table is preserved.

The syntax for the TRUNCATE TABLE statement in MySQL is:
TRUNCATE TABLE table_name;

Delete:
The DELETE statement in SQL is a Data Manipulation Language(DML) Command. It is used to delete existing records from an existing table. We can delete a single record or multiple records depending on the condition specified in the query.

The syntax for the DELETE statement in SQL is:
DELETE FROM table_name [WHERE conditions];

Drop:
DROP statement is a Data Definition Language(DDL) Command which is used to delete existing database objects. It can be used to delete databases, tables, views, triggers, etc.
The syntax for the DROP statement in SQL is:

The syntax for the DELETE statement in SQL is:
DELETE FROM table_name [WHERE conditions];

18) Find out the range of data types

Data Type	Lower Range	Upper Range
Bit	0	1
tinyint	0	255
Smallint	-2^15 (-32,768)	2^15-1 (32,767)
Int	−2^31 (−2,147, 483,648)	2^31−1 (−2,147, 483,647
Bigint	−2^63 (−9,223,372, 036,854,775,808)	2^63−1 (−9,223,372, 036,854,775,807)
Decimal	−10^38+1	10^381−1
Numeric	−10^38+1	10^381−1
Smallmoney	-214,478.3648	+214,478.3647
Money	−922,337, 203, 685,477.5808	+922,337, 203, 685,477.5807

17)One line definition for below:

SQL Server Editions-

1) Enterprise

2) Standard

3) Developer

4) Express

Enterprise
SQL Server Enterprise Edition includes both the core database engine and add-on services, with a range of tools for creating and managing a SQL Server cluster. It can manage databases as large as 524 petabytes and address 12 terabytes of memory and supports 640 logical processors (CPU cores).

Standard
SQL Server Standard edition includes the core database engine, along with the stand-alone services. It differs from Enterprise edition in that it supports fewer active instances (number of nodes in a cluster) and does not include some high-availability functions such as hot-add memory (allowing memory to be added while the server is still running), and parallel indexes.

Developer
SQL Server Developer Edition includes the same features as SQL Server Enterprise Edition, but is limited by the license to be only used as a development and test system, and not as production server. Starting early 2016, Microsoft made this edition free of charge to the public.

Express
SQL Server Express Edition is a scaled down, free edition of SQL Server, which includes the core database engine. While there are no limitations on the number of databases or users supported, it is limited to using one processor, 1 GB memory and 10 GB database files.