

UNIT II

1. Define variable.

Variables are the named areas in the memory used to store values that may be changed during the execution of the program. Therefore variable is defined as a quantity whose value changes during program execution.

Example: x = 100;
 y = "John";

Here the variable x is used to store the value 100 and the variable y is used to store the value John. If the value of a variable is a string, it must be enclosed within double quotes.

The value of the variable can be changed any time by using the assignment statement.

Example: x = 100;
 x = x + 50;

2. Define constant.

Constants are used to hold values that cannot be changed during the execution of the program. Constants are processed faster than variables. The general syntax to declare constants is

const datatype constant name = Value;

Where const. - Keyword
 contant name - User defined name
 datatype - Type of the constant

Example: Const double Pi = 3.14;

The value 3.14 assigned to Pi cannot be changed during the execution. That is Pi = Pi + 100.8 is not possible.

3. How constants are declared in C#?

const datatype constant-name = value;

Where const. - Keyword
 contant name - User defined name
 datatype - Type of the constant

Example: Const double Pi = 3.14;

4. List the relational operators in C#.

Operator	Purpose	Example	Result
==	Check if a value is equal to another	Let a=50 b=(a==55);	False
!=	Check if a value is not equal to another	b=(a!=55);	True
>	Check if a value is greater than another	b=(a>55);	False
<	Check if a value is less than another	b=(a<55);	True
>=	Check if a value is greater than or equal to another	b=(a>=35);	True
<=	Check if a value is less than or equal to another	b=(a<=55);	True

5. Write the C# statement to find 50 mod 7.

```
int res;  
res=50 % 7;  
after execution res=1.
```

6. Write the syntax of if ... else statement in C#.

```
if (condition)
{
Statement block-1;
}
else
{
Statement block-2;
}
```

7. What is loop?

Loop statements are used to execute a set of statements repeatedly.

8. Write the syntax of do loop in C#.

```
do
{
Statement(s);
}
while (condition);
```

9. Write the syntax to declare a 2D array in c#. Or How 2D array are declared in C#?

Give an example.

`datatype[,] arrayname=new datatype[row size, colume size];`

Example : `int [,]a=new int[3, 4];`

10. List any 5 datatypes in C#.NET.

int, float, double, char, bool

11. Explain decision making statements in C#.NET.

Decision statements in C#.NET are used to control the flow of a program by checking one or more conditions. There are three types of decision statements in C#.NET.

- 1. Simple if 2. If ... Else 3. Switch ... case**

12. Define jagged array.

If each element of an array is another array, then it is called Jagged array. In other words, Jagged array is an **array of arrays**.

13. Define overriding.

Method Overriding is a technique that allows the invoking of functions from another class (base class) in the derived class. Creating a method in the derived class with the same name as a method in the base class is called as method overriding.

14. Define constructor.

Instantiate means giving initial values to the instance variables of an object. It is done by a special method called constructors. Constructor is automatically called at the time of object creation..

15. Give examples for reference data types.

Array, string, class.

16. What do you mean instantiate an object?

Instantiate means giving initial values to the instance variables of an object.

17. What do you mean value data types?

A Value type variable holds the data within its own memory location and own copy of data for operations. The operations on a value type variable do not affect the others. char, date, bool, structures, enumerated, int, float are the important value type variables.

18. Define class.

Class is a user defined data type that contains a group of related functions and variables. The functions are known as methods and variables are known as fields. Object is a variable of type class and used to access the class. The fields and methods are known as members of the class.

19. Define object.

The data members and member functions of a class can be accessed in a program only by creating an object for the class. Object is a variable of type class. We can create any number of objects for a class.

20. Write an example how a method is accessed using object.

```
class Rectangle
{
    double height;
    double length;
    Rectangle()
    {
        height=20;
        length=30;
    }
    double area()
    {
        return(height * length);
    }
}
```

In the above code height and length are the instance variables of the class rectangle. The initial values for height and length are given using the constructor.

Now we can create object to this class and call the methods of the class like.

```
Rectangle r1 = new Rectangle ();
double a = r1. area();
```

Here area () is a member function of the class Rectangle.

21. Write the syntax for creating object.

```
class-name object1, object2, ... objectn;
object = new class-name();
```

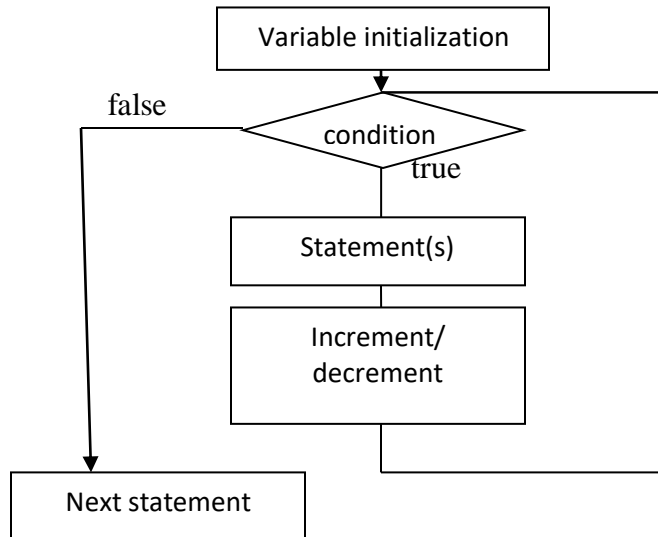
22. Write the syntax to access a method of a class.

```
object.method_name();
```

23. What is the use of switch... case statement.

When large number of comparisons are required on a single expression, the If statements become complex and need large coding. To avoid this problem, switch statement is used. This statement evaluates a single expression and compares it with a list of values. If a match found, then the on corresponding block is executed. If no match found then the default block is executed.

24. Draw the flow diagram of for loop.



25. Write about arithmetic operators in C#.

Arithmetic operators are used to perform simple arithmetic calculations such as addition and subtraction.

Operator	Purpose	Example Let a=50	Output
+	Addition	B=a+5;	55
-	Subtraction	B=a-5;	45
*	Multiplication	B=a*10;	500
/	Division	B=a/2	25
%	Modulus (remainder after division)	B=a%7;	1
++	Increment	A++;	51
--	Decrement	a--;	49

26. Write the C# code to find 1+2+....+n using for loop.

```

int n,i,sum=0;
n=Convert.ToInt32(Console.ReadLine());
for(i=1;i<=n;i++)
sum=sum+i;
Console.WriteLine("Sum="+sum);
  
```

27. Give an example for enumeration.

Example 1 : enum days{ sun, mon, tue, wed, thu, fri, sat };

Output : sun=0, mon=1, tue=2 , wed=3, thu=4, fri=5, sat=6

Example 2 : enum colors { white, red=7, blue, green, black=6};

Output : white=0, red=7, blue=8, green=9, black=6

28. What is the use of NOT operator?

This logical operator is used to perform a **logical negation** on a Boolean expression.

Syntax : result = !(Boolean expression);

The given table shows the result of NOT operation.

Boolean Expression	Result
True	False
False	True

29. Draw the truth table for XOR.

Expression 1	Expression2	Expression1 XOR Expression2
T	T	T
T	F	F
F	T	F
F	F	T

30. Define structure.

Structure is an user defined data type used to store more than one data of different type. The data items are called members and can be accessed individually. Structure cannot be declared within a function.

Syntax :

```

struct structure-name
{
    declaration of members;
}

```

31. Define constructor. Write the syntax of a constructor.

Instantiate means giving initial values to the instance variables of an object. It is done by a special method called constructors. Constructor is automatically called at the time of object creation.

Syntax:

```

Constructor-name (arguments)
{
    Initial values;
}

```

32. Write the rules for constructor creation.

- Name of the constructor is same as the name of the class.
- Constructor has no return type.
- Arguments may be empty.
- If it has arguments, the value of the arguments must be given in the object declaration.

33. Define value data types and reference data types.

A Value type variable holds the data within its own memory location and own copy of data for operations. The operations on a value type variable do not affect the others. char, date, boolean, structures, enumerated, int, float are the important value type variables.

A reference type variable contains a pointer (address) to a memory location that holds the data. More than one reference type variables can refer to the same object. So the operation on one variable affects the other.

34. Write the general syntax of a class.

```
class class-name
{
    datatype variable1;
    datatype variable2;
    ---
    datatype variable n;
    datatype method-name-1()
    {
        body of the method
    }
    ---
    datatype method-name-n()
    {
        body of the method
    }
}
```

35. Write the C# code to create a class student with a member function and three data members.

```
class student
{
    int regno;
    int mark1;
    int mark1;
    int total()
    {
        int t;
        t=mark1+mark2;
        return(t);
    }
}
```

36. What is the use of 'this' keyword. Or What is the use of 'this' reference?

this is an implicit pointer to every method in a class. This pointer contains the address of the object which calls the method.

37. Write an example program for 'this' keyword.

```
class demo
{
    int x=10;
    int y=20;
```

```

public void get()
{
    x=500;
    y=700;
    Console.WriteLine("x={0} y={1}",x, y);
    Console.WriteLine ("x={0} y={1}",this.x, this.y);
}
}

```

Output :

500 700 (display local values)
10 20 (display field values ie. this.x and this.y)

38. Differentiate while loop and do loop.

	while loop	do-while loop
1	Syntax : while (condition) { statement(s); }	Syntax : do { statement(s); } while (condition);
2	Entry controlled	Exit controlled
3	While loop statement is used to execute a set of statements repeatedly as long as a given condition is true.	Do loop is used to execute a set of statements repeatedly when a given condition is true.
4	The loop is executed only when the condition is true. When the condition becomes false, program control passes to the line immediately following the loop.	Do loop checks its condition at the end of the loop, So it is guaranteed to execute the statement block at least one time.
5	Example : i=1; while(i<=10) { Console.WriteLine("welcome"); i++; }	Example : i=1; do { Console.WriteLine("welcome"); i++; } while(i<=10);
6	<pre> graph TD Start([Start]) --> Eval{Evaluate Condition} Eval -- true --> Exec[Execute Statements] Exec --> Eval Eval -- false --> Stop([Stop]) </pre>	<pre> graph TD Start([Start]) --> Body[Code Inside body of do-while loop] Body --> Cond{Condition} Cond -- True --> Body Cond -- False --> End([End of Loop]) </pre>

39. What is the difference between class and interface?

S.NO	Class	Interface
1.	<p>Syntax : class class-name { datatype variable1; datatype variable2; --- datatype variable n; datatype method-name1() { body of the method } --- datatype method-namen() { body of the method } }</p>	<p>Syntax : interface interface -name { datatype variable1; datatype variable2; --- datatype variable n; datatype method-name1(parameter list); --- datatype method-namen(parameter list); } }</p>
2.	<p>Class is a user-defined data type that allows the grouping of data members, methods, properties, and events. A class is nothing but a blueprint that defines data and behaviour. Objects are instances of the class.</p>	<p>An Interface contains only the signature of members: methods, properties, events or indexers. It does not contain the definition of these members. It is up to the class which is deriving this interface to write the definition of these members. It is mandatory for a class to implement all the members of the interface.</p>
3.	<p>A class can be instantiate.</p>	<p>Interface cannot be instantiated.</p>
4.	<p>A class may contain abstract methods, concrete methods.</p>	<p>An interface contains only abstract methods.</p>
5.	<p>Members of a class can be public, private, protected or default.</p>	<p>All the members of the interface are public by default.</p>

UNIT II

PART – A

1. How constants are created in C#.NET?
2. Explain Decision Making Statements in C#.NET.
3. What is jagged array?
4. Define Structure with syntax.
5. List any five data types in C#.
6. Define Checkbox.

PART – B

1. What is the difference between class and interface?
2. What is the use of this reference?
3. Differentiate foreach statement with for.
4. Define value data type and reference data type.
5. What is the use of NOT operator?
6. Differentiate show() and showdialog().
7. Explain any three operators.

PART - C

1. Explain the various Looping Statements in c#.NET with example. (10)
2. List the various types of operators available in C#.NET. Explain. (10)
3. Discuss the different types of arrays in C#.NET with suitable examples. (5)
4. Write short notes on Abstract and Override methods. (5)
5. Define Array. Differentiate regular and jagged arrays. (5)
6. Write short notes on structures and enumerations. (5)
7. Define class. Explain how objects are created in C#? (5)
8. What is instantiating the object? Explain the types of constructors with example. (5)
9. Explain single dimensional array and dynamic array with an example. (5)
10. Write an program in C#.NET for adding 'n' natural numbers using class and object. (5)