

## UNIT II

### 1. Define variable.

Variable என்பது பெயரிடப்பட்ட memory பகுதியாகும். ஒவ்வொரு variable-ம் ஒரு name மற்றும் ஒரு மதிப்பை கொண்டிருக்கும். Variable-ன் மதிப்பு program run ஆகும் போது மாற்றமடையலாம்.

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

- x,y என்பவை variable-களாகும்
- 100, "CBT" என்பவை அவற்றின் மதிப்பு ஆகும்.

### 2. Define constant.

Constant என்பது Variable-ஐ போன்றே ஒரு மதிப்பை கொண்டிருக்கும். ஆனால் இந்த மதிப்பை program execution நேரத்தில் மாற்ற முடியாது.

### 3. How constants are declared in C#?

const datatype constant-name = value;

where const            – keyword  
          constant-name – user இட்டபெயர்  
          datatype       – constant-ன் type

Example : const double PI=3.14;

### 4. List the relational operators in C#.NET.

Operator	Purpose	Example	Result
		Let a=50	
==	Check if a value is equal to another	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#.NET.

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

**7. What is loop?**

குறிப்பிட்ட சில statement-களை பல தடவைகள் மீண்டும் மீண்டும் execute செய்வதற்கு loop என்று பெயர்.

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

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

**9. Write the syntax to declare a 2D array in C#.NET. or How 2D array are declared in C#.NET? 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 making statement-கள் சில condition-களை சோதித்துவிட்டு அதன் அடிப்படையில் program-இன் flow-வை மாற்ற பயன்படுகின்றது. C#-இல் மூன்று Decision making statement-கள் பயன்படுத்தப்படுகின்றன. அவை

1. Simple if
2. if ... else
3. switch ... case

**12. Define jagged array.**

ஒரு array-இல் உள்ள ஒவ்வொரு உறுப்பும் இன்னொரு array ஆக இருந்தால் அதற்கு jagged array என்று பெயர்.

**13. Define overriding.**

Overriding என்பது super class-இல் ஏற்கனவே define செய்யப்பட்ட method-களை subclass-களில் அதே argument-களுடன் redefine செய்யக்கூடிய process ஆகும்.

**14. Define constructor.**

ஒரு object-ன் instance variable-களுக்கு initial value கொடுப்பது instantiate எனப்படும். இதற்கு constructor என்னும் சிறப்பு method-கள் பயன்படுகின்றன.

**15. Give examples for reference data types.**

Array, string, class.

**16. What do you mean instantiate an object?**

ஒரு Object-ன் instance variable-களுக்கு initial value கொடுப்பது instantiate an object எனப்படும்.

### 17. What do you mean value data types?

Value type variable-கள் தங்களது data-வை அவற்றுக்கு சொந்தமான memory location-இல் வைத்திருப்பதோடு அந்த data-வை operate செய்வதற்காக அதன் copy-யை கொண்டிருக்கும்.

### 18. Define class.

ஒன்றுக்கொன்று தொடர்புடைய பல function-களையும் variable-களையும் உள்ளடங்கிய ஒரு user defined datatype-க்கு class என்று பெயர். Class-இல் உள்ள function-களுக்கு methods என்றும், variable-களுக்கு fields என்றும் பெயர்.

### 19. Define object.

Object என்பது ஒரு class-ஐ தனது datatype-ஆக கொண்ட ஒரு Variable ஆகும். object மூலமாக class-இல் உள்ள method-களும், field-களும் access செய்யப்படுகின்றன. Methods, மற்றும் field-கள் class-ன் member-கள் எனப்படுகின்றன.

### 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);
    }
}
```

இந்த class-க்கு இப்போது ஒரு object-ஐ உருவாக்கி அதன் method-களை call செய்யலாம்.

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

### 21. Write the syantx for creating object.

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

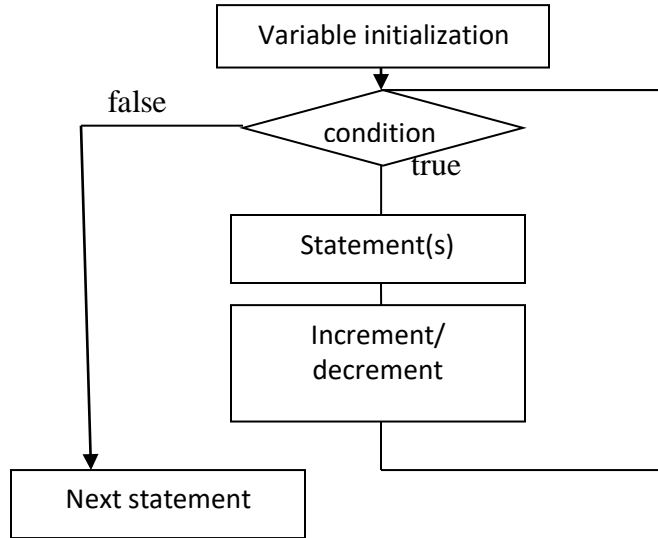
### 22. Write the syntax to access a method of a class.

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

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

switch... case statement-ஐ பயன்படுத்தி ஒரே ஒரு expression-ன் மீது பல comparison-களை செய்யலாம்.

24. Draw the flow diagram of for loop.



25. Write about arithmetic operators in C#.NET.

கூட்டல், கழித்தல் போன்ற எளிய arithmetic கணக்கீடுகளை செய்வதற்கு இவ்வகை operator-கள் பயன்படுகின்றன.

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#.NET 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?**

ஒரு Boolean expression மீது logical negation செய்ய இந்த operator பயன்படுகிறது.

**Syntax :** result = !(Boolean expression);

NOT operation-ன் result-ஐ கீழ்க்கண்ட table காட்டுகிறது.

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.**

ஒன்று அல்லது அதற்கு மேற்பட்ட வெவ்வேறு datatype-களை தொகுத்து ஒரு தனி datatype ஆக declare செய்தால் அதற்கு structure என்று பெயர்.

**Syntax :** struct structure-name  
{  
Declaration of members;  
}

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

ஒரு Object-ன் instance variable-களுக்கு initial value கொடுப்பது instantiate an object எனப்படும். இதற்கு constructor என்னும் சிறப்பு method-கள் பயன்படுகின்றன.

**Syntax :** constructor -name(arguments)  
{  
Initial values;  
}

### 32. Write the rules for constructor creation.

1. Class-ன் பெயரில்தான் constructor இருக்கும்.
2. Constructor-க்கு return type கிடையாது.
3. Argument-கள் empty ஆக இருக்கலாம்.
4. Argument-கள் இருந்தால் அவற்றின் மதிப்பு object declaration-ல் கொடுக்கப்படவேண்டும்.

### 33. Define value data types and reference data types.

Value type variable-கள் தங்களது data-வை அவற்றுக்கு சொந்தமான memory location-இல் வைத்திருப்பதோடு அந்த data-வை operate செய்வதற்காக அதன் copy-யை கொண்டிருக்கும்.

Reference type variable-கள் நேரடியாக data-வை வைத்திருப்பதில்லை. மாறாக அவை ஒரு pointer-ஐ கொண்டிருக்கும். அந்த pointer எந்த memory location-ஐ குறிக்கிறதோ அங்கு data இருக்கும்.

### 34. Write the general syntax of a class.

```
class class-name
{
    datatype variable1;
    datatype variable2;
    ---
    datatype variable n;
    datatype method-name1()
    {
        body of the method
    }
    ---
    datatype method-namen()
    {
        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” என்பது ஒரு class-ல் உள்ள ஒவ்வொரு method-க்கும் உரிய implicit pointer ஆகும். Class-ல் உள்ள method-ஐ call செய்கிற object-ன் address—ஐ இந்த pointer கொண்டிருக்கும்.

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

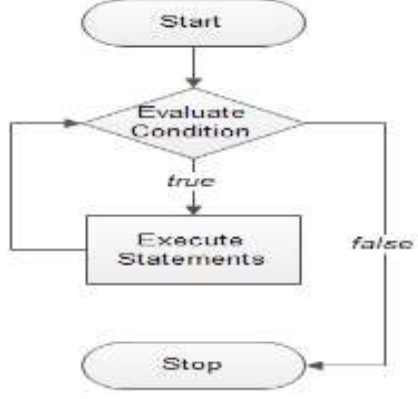
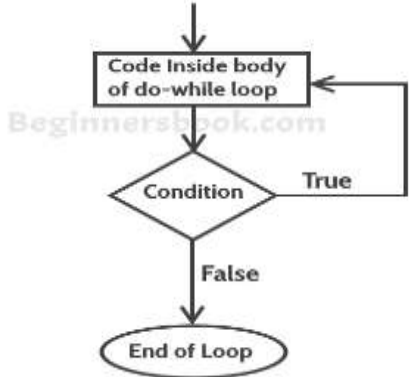
```
class demo
{
    int x=10;
    int y=20;
    public void get()
    {
        x=500;
        y=700;
        Console.WriteLine("x={0} y={1}",x,y);
        Console.WriteLine("x={0} y={1}",this.x,this.y);
    }
}
```

**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.	<b>Syntax :</b> while (condition) { statement(s); }	<b>Syntax :</b> do { statement(s); } while (condition);
2.	<b>Entry controlled</b>	<b>Exit controlled</b>
3.	குறிப்பிட்ட சில statement-களை, ஒரு condition ஆனது true ஆக இருக்கும் வரை மீண்டும் மீண்டும் execute செய்யும்.	while statement-ஐ போலவே குறிப்பிட்ட சில statement-களை, ஒரு condition ஆனது true ஆக இருக்கும் வரை மீண்டும் மீண்டும் execute செய்யும்.
4.	Condition ஆனது statement block-க்கு முன்பாய் சோதிக்கப்படுவதால், condition false ஆக இருந்தால் ஒருமுறை loop execute செய்யாது.	Condition ஆனது statement block-க்கு பிறகு சோதிக்கப் படுவதால் குறைந்தபட்சம் ஒரு தடவையாவது loop execute ஆகிறது.

5.	<b>Example :</b> <pre>i=1; While(i&lt;=10) { Console.WriteLine("welcome"); i++; }</pre>	<b>Example :</b> <pre>i=1; do { Console.WriteLine("welcome"); i++; } while(i&lt;=10);</pre>
6	 <pre> graph TD     Start([Start]) --&gt; Eval{Evaluate Condition}     Eval -- true --&gt; Exec[Execute Statements]     Exec --&gt; Eval     Eval -- false --&gt; Stop([Stop]) </pre>	 <pre> graph TD     Start([Start]) --&gt; Body[Code Inside body of do-while loop]     Body --&gt; Cond{Condition}     Cond -- True --&gt; Body     Cond -- False --&gt; End([End of Loop]) </pre>

**39. What is the difference between class and interface?**

S.NO	Class	Interface
1.	<b>Syntax :</b> <pre>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     } }</pre>	<b>Syntax :</b> <pre>interface interface -name {     datatype variable1;     datatype variable2;     ---     datatype variable n;     datatype method-name1(parameter list);     ---     datatype method-namen(parameter list); }</pre>
2.	<p>ஒன்றுக்கொன்று தொடர்புடைய பல function-களையும் variable-களையும் உள்ளடங்கிய ஒரு user defined datatype-க்கு class என்று பெயர். Class-இல் உள்ள function-களுக்கு methods என்றும், variable-களுக்கு fields என்றும் பெயர்</p>	<p>interface-இல் methods-கள், properties-கள் மற்றும் events-களின் declaration-ஐ மட்டுமே கொண்டுள்ளது. ஆனால் implementation-ஐ கொண்டிருக்கவில்லை.</p>



3.	A class can be instantiate.	Interface cannot be instantiated.
4	class members private ஆகவோ or public ஆகவோ இருக்கலாம்.	interface-இல் private members-ஐ சேர்க்க முடியாது. அனைத்து member-களும் public by default.

## 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 datatypes 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 datatype and reference datatype.
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.
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#?
8. What is instantiating the object? Explain the types of constructors with example.
9. Explain single dimensional array and dynamic array with an example.
10. Write an program in C#.NET for adding 'n' natural numbers using class and object.