	TOPIC	C	CPP	JAVA	C#.NET	VB.NET
INTEGERS	8 bit (byte)	signed char unsigned char	signed char unsigned char	byte	byte sbyte	Byte SByte
	16 bit (short integer)	short unsigned short	short unsigned short	short char	short ushort	Short Ushort
	32 bit	long unsigned long	long unsigned long	int	int uint	Integer UInteger
	Word size	int unsigned int	int unsigned int	N/A	N/A	N/A
	Arbitrarily precise (bignum)	N/A	N/A	java.math.BigInteger	System.Numerics.BigInteger (.NET 4.0)	System.Numerics.BigInteger (.NET 4.0)
Floating point	Single Precision	float	float	float	float	Single
	Double Precision	double	double	double	double	Double
Other variable type	Character	char	char	char	char	Char
	String	N/A	string	String	String	String
	Boolean	bool	bool	boolean	bool	Boolean
	Enumeration	enum name {item1, item2,…};	enum name {item1, item2,…};	enum name {item1, item2,…}	enum name {item1, item2}	Enum name Item1 End Enum
	object Universal	void *	void *	Object	object	Object
Other type	Records Structure	struct name {type name;};	N/A	N/A	struct name {type name;};	Structure name Dim a As type End Structure
	Unions	union name {type name;…};			N/A	N/A

<< >> quoted text is optional ■ Author : Kamini Chaudhari, Harshal Chaudhari, Sachin Sonawane.

Sponsored by
Khandesh College Education Society's
M. J. College Campus, Jilha peth, Jalgaon - 425001 (M.S.) India
Tel/Fax: +91 - 257 - 2235646, Web : http://www.isdntc.com
ISDNTC
Institute of Software Development & Training Centre

	TOPIC	C	CPP	JAVA	C#.NET	VB.NET
Declaration	variable	Type_name name << = initial_value >>;	Type_name name << = initial_value >>;	Type_name name << = initial_value >>;	Type_name name << = initial_value >>;	Dim name As Type << = initial_value >>
	constant Type synonym	Enum{name = value }; typedef type synonym;	const Type_name name= value; typedef type synonym;	final Type_name name= value; N/A	const Type_name name= value; using synonym = type;	Const name As Type = value Imports synonym = type
Control flow	if	if(condition){instruction;} << else {instruction}>>	if(condition){instruction;} << else {instruction;}>>	if(condition){instruction;} << else {instruction;}>>	if(condition){instruction;} << else {instruction;}>>	If condition Then` instruction << Else instruction >> End If

	else if	if(condition)
		{instruction;}
		else if(condition)
		{instruction;}
		<< else
		{instruction;} >>
			if(condition)
			{instruction;}
			else if(condition)
			{instruction;}
			<< else
			{instruction;} >>
				if(condition)
				{instruction;}
				else if(condition)
				{instruction;}
				<< else
				{instruction;} >>
					if(condition)
					{instruction;}
					else if(condition)
					{instruction;}
					<< else
					{instruction;} >>
						If condition Then
						instruction
						ElseIf condition Then
						<< Else
						Instruction >>
						End If

	Select case	switch (variable)
		{
		case case1: instruction;
		<<break;>>
		<< default: instruction;
		>>
		}
			switch (variable)
			{
			case case1: instruction;
			<<break;>>
			<< default: instruction;
			>>
			}
				switch (variable)
				{
				case case1: instruction;
				<<break;>>
				<< default: instruction;
				>>
				}
					switch (variable)
					{
					case case1: instruction;
					<<break;>>
					<< default: instruction;
					>>
					}
						Select Case variable
						Case case1
						instruction
						<< Case Else
						instruction >>
						End Select

	Conditional Expression	if (Condition ? Value_True : Value_False)	if (Condition ? Value_True : Value_False)	if (Condition ? Value_True : Value_False)	if (Condition ? Value_True : Value_False)	iff (Condition , Value_True , Value_False)

Loops	while	while(condition)
		{
		instruction;
		}
			while(condition)
			{
			instruction;
			}
				while(condition)
				{
				instruction;
				}
					while(condition)
					{
					instruction;
					}
						do until condition
						instructions
						Loop or
						While condition
						instructions
						End While

<< >> quoted text is optional

 Author : Kamini Chaudhari, Harshal Chaudhari, Sachin Sonawane.

Sponsored by
Khandesh College Education Society's

M. J. College Campus, Jilha peth, Jalgaon - 425001 (M.S.) India
Tel/Fax: +91 - 257 - 2235646, Web : http://www.isdntc.com

ISDNTC
Institute of Software Development & Training Centre

	TOPIC	C	CPP	JAVA	C#.NET	VB.NET
Declaration	Do while	do { instruction; } while(condition);	do { instruction; } while(condition);	do { instruction; } while(condition);	do { instruction; } while(condition);	do instructions Loop While condition or do instructions Loop Until Condition

	for	for(<<type>>i=first; i<last; i++)
		{
		instruction;
		}
			for(<<type>>i=first; i<last; i++)
			{
			instruction;
			}
				for(<<type>>i=first; i<last; i++)
				{
				instruction;
				}
					for(<<type>>i=first; i<last; i++)
					{
					instruction;
					}
						For i =first to last <<Step 1>>
						instruction
						Next I

	foreach	N/A	for_each(start, end, function)	for(type item:set)
				{
				instruction;
				}
					foreach (type item in set)
					{
					instruction;
					}
						for each item as Type in set
						instruction
						Next item

	throw		throw exception;	throw exception;	throw exception;	throw exception
	handler		try {instructions } catch «(exception)» {instructions } << finally {instructions }>>	try {instructions } catch <<(exception)>> {instructions } << finally {instructions }>>	try {instructions } catch <<(exception)>> {instructions } << finally {instructions }>>	try instruction Catch ex As Exception instruction Finally instruction End Try
	assertion	assert(condition);	assert(condition);	assert(condition);	System.Diagnostics. Debug.Assert(Condition);	System.Diagnostics. Debug.Assert(Condition);
Control flow	Exit block	break;	break;	break;	break;	Exit block
	continue	continue;	continue;	continue;	continue;	continue block
	Label	label:	label:	label:	label:	label:
	GOTO	goto label;	goto label;	goto label;	goto label;	Goto label

<< >> quoted text is optional

 Author : Kamini Chaudhari, Harshal Chaudhari, Sachin Sonawane.

Sponsored by
Khandesh College Education Society's
M. J. College Campus, Jilha peth, Jalgaon - 425001 (M.S.) India
Tel/Fax: +91 - 257 - 2235646, Web : http://www.isdntc.com
ISDNTC
Institute of Software Development & Training Centre

	TOPIC	C	CPP	JAVA	C#.NET	VB.NET
function	Calling function	foo(<<parameters>>);	foo(<<parameters>>);	foo(<<parameters>>);	foo(<<parameters>>);	foo(<<parameters>>)
	Basic / Void Function	void foo(<<parameters>>) {instructions;}	void foo(<<parameters>>) {instructions;}	void foo(<<parameters>>) {instructions;}	void foo(<<parameters>>) {instructions;}	sub foo(<<parameters>>) instructions End sub
	Value returning function	type foo(<<parameters>>) {instructions; return value; }	type foo(<<parameters>>) {instructions; return value; }	type foo(<<parameters>>) {instructions; return value; }	type foo(<<parameters>>) {instructions; return value; }	Function foo(<<parameters>>) As type instructions return value End Function
	Required Main Function	<<global declarations>> int main(<<int argc,char *argv[]>>) {instructions;}	<<global declarations>> int main(<<int argc,char *argv[]>>) {instructions;}	public static void main(string []args) {instructions;}	static void main(<<string []args) {instructions;} or static int main(<<string []args>>) {instructions;}	sub main(<<byval cmdarg() As string >>) instructions End sub or Function main(<<byval cmdarg() As string >>) instructions End Function
Type conversion	String to integer	Integer = atoi (string);	istringstream(string) >> number;	Integer=Integer.parseInt(string);	int = int.Parse(string);	Integer = CInt(string)
	String to long integer	Integer = atol (string);	istringstream(string) >>	long=Long.parseLong(string);	long = long.Parse(string);	long = CLng(string)
	String to floating point	Integer = atof (string);	istringstream(string) >> number;	float=Float.parseFloat(string); double=Double.parseDouble(string);	float = float.Parse (string) double = double.Parse(ptring)	single = CSng(string) double = CDbl(string
	Integer to string	sprintf(string,"%i",integer);	ostringstream o; o << number; string = o.str();	String=Integer.toString(integer);	String=number.toString();	String = CStr (number)
	Floating point to string	sprintf(string,"%f",float);	ostringstream o; o << number; string = o.str();	String=Float.toString(float); String=Double.toString(double);	String=number.toString();	String = CStr (number)

<< >> quoted text is optional ■ Author : Kamini Chaudhari, Harshal Chaudhari, Sachin Sonawane.

Sponsored by
Khandesh College Education Society's

M. J. College Campus, Jilha peth, Jalgaon - 425001 (M.S.) India
Tel/Fax: +91 - 257 - 2235646, Web : http://www.isdntc.com

ISDNTC
Institute of Software Development & Training Centre

	TOPIC	C	CPP	JAVA	C#.NET	VB.NET

Standard input output	Read from	`scanf(format, &x);`
		`fscanf(stdin, format, &x);`
			`cin >> x;`
			`getline(«std::»cin, str);`
				`x = System.in.read(); or`
				`x = new Scanner(System.in).nextInt();`
				or
				`x = new Scanner(System.in).nextLine();`
					`x = Console.Read();`
					or
					`x = Console.ReadLine();`
						`x = Console.Read()`
						or
						`x = Console.ReadLine()`

	Write stdout	`printf(format, x);`
		`fprintf(stdin, format, &x);`
			`cout << x;`	`System.out.print(x);`
				`System.out.printf(format,x);`
				`System.out.println(x);`
					`Console.Write(<<format, >>x);`
					or
					`Console.WriteLine(<<format,>>x);`
						`Console.Write(<<format,>>x)`
						or
						`Console.WriteLine(<<format,>>x)`

	Write Error	`fprintf(stderr,format,x);`	`cerr << x;`
			`clog << x;`
				`System.err.print(x);`
				`System.err.printf(format,x); or`
				`System.err.println(x);`
					`Console.Error.Write(<<format, >>x);`
					or
					`Console.Error.WriteLine(<<format,>>x);`
						`Console.Error.Write(<<format,>>x)`
						or
						`Console.Error.WriteLine(<<format,>>x)`

Commandline argument	Argument Values	`argv[n]`	`argv[n]`	`argv[n]`	`argv[n]`	`cmdArgs(n)`
	Argument count	`argc`	`argc`	`args.Length`	`args.Length`	`cmdArgs.Length`
	Program Name	`first Argument`	`first Argument`	`first Argument`	`Assembly.GetEntryAssembly().Location;`	`[Assembly].GetEntryAssembly().Location`

Execution of commands	Shell Command	`System("Command");`	`System("Command");`	`Runtime.exec(command);`
				or
				`new ProcessBuilder(command).start();`
						`Microsoft.VisualBasic.Interaction.Shell(command <<, WindowStyle>> <<, isWaitOnReturn>>)`

	Execute Program	`execl(path,args);` `execv(path,arglist);`	`execl(path,args);` `execv(path,arglist);`		`System.Diagnostics.Process.Start(path, argstring);`	`System.Diagnostics.Process.Start(path, argstring)`

<< >> quoted text is optional ■ Author : Kamini Chaudhari, Harshal Chaudhari, Sachin Sonawane.

Sponsored by
Khandesh College Education Society's
M. J. College Campus, Jilha peth, Jalgaon - 425001 (M.S.) India
Tel/Fax: +91 - 257 - 2235646, Web : http://www.isdntc.com
ISDNTC
Institute of Software Development & Training Centre