

ROHINI College of Engineering and technology

TRP1601- SOFT SKILLS

UNIT-I INTRODUCTION TO C & C++

OVERVIEW OF C

- ❖ C is a general-purpose, high-level language that was originally developed by Dennis M. Ritchie to develop the UNIX operating system at Bell Labs. C was originally first implemented on the DEC PDP-11 computer in 1972.
- ❖ In 1978, Brian Kernighan and Dennis Ritchie produced the first publicly available description of C, now known as the K&R standard.
- ❖ The UNIX operating system, the C compiler, and essentially all UNIX application programs have been written in C. C has now become a widely used professional language for various reasons –
 - Easy to learn
 - Structured language
 - It produces efficient programs
 - It can handle low-level activities
 - It can be compiled on a variety of computer platforms.

CONSTANT:

A constant is a value or variable that can't be changed in the program, for example: 10, 20, 'a', 3.4, "c programming" etc.

There are different types of constants in C programming.

Constant	Example
Decimal Constant	10, 20, 450 etc.
Real or Floating-point Constant	10.3, 20.2, 450.6 etc.
Octal Constant	021, 033, 046 etc.
Hexadecimal Constant	0x2a, 0x7b, 0xaa etc.
Character Constant	'a', 'b', 'x' etc.
String Constant	"c", "c program", "c in rcet" etc.

There are two ways to define constant in C programming.

1. const keyword
2. #define preprocessor

1) C const keyword

The const keyword is used to define constant in C programming.

```
const float PI=3.14;
```

Now, the value of PI variable can't be changed.

Example:

```
1. #include<stdio.h>
2. int main(){
3.     const float PI=3.14;
4.     printf("The value of PI is: %f",PI);
5.     return 0;
6. }
```

Output:

The value of PI is: 3.140000

2) C #define preprocessor

The #define preprocessor is also used to define constant. We will learn about #define preprocessor directive later.

Variables:

A variable is a name of the memory location. It is used to store data. Its value can be changed, and it can be reused many times.

It is a way to represent memory location through symbol so that it can be easily identified.

Let's see the syntax to declare a variable:

```
type variable_list;
1. int a;
2. float b;
3. char c;
```

Types of Variables in C:

There are many types of variables in c:

1. local variable
2. global variable
3. static variable
4. automatic variable
5. external variable.

Local Variable

A variable that is declared inside the function or block is called a local variable.

It must be declared at the start of the block.

```
void function1(){  
    1. int x=10;//local variable  
    2. }
```

Global Variable

A variable that is declared outside the function or block is called a global variable. Any function can change the value of the global variable. It is available to all the functions.

It must be declared at the start of the block.

```
1. int value=20;//global variable  
2. void function1(){  
3. int x=10;//local variable  
4. }
```

Static Variable

A variable that is declared with the static keyword is called static variable.

It retains its value between multiple function calls.

```
1. void function1(){  
2. int x=10;//local variable  
3. static int y=10;//static variable  
4. x=x+1;  
5. y=y+1;  
6. printf("%d,%d",x,y);  
7. }
```

Automatic Variable

All variables in C that are declared inside the block, are automatic variables by default. We can explicitly declare an automatic variable using auto keyword.

```
1. void main(){  
2. int x=10;//local variable (also automatic)  
3. auto int y=20;//automatic variable  
4. }
```

External Variable

We can share a variable in multiple C source files by using an external variable. To declare an external variable, you need to use extern keyword.

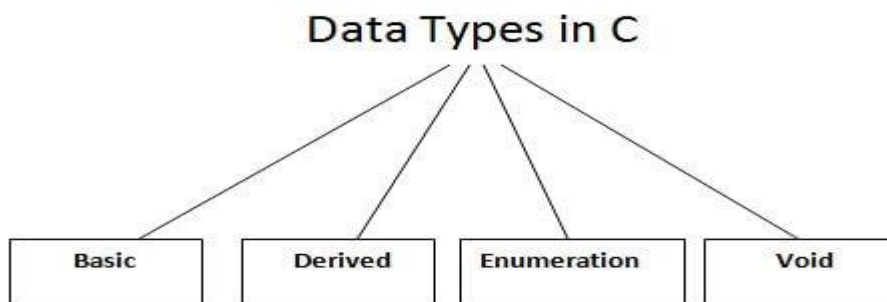
myfile.h

```
extern int x=10;//external variable (also global)
```

1. #include "myfile.h"
2. #include <stdio.h>
3. void printValue(){
4. printf("Global variable: %d", global_variable);
5. }

DATA TYPES

A data type specifies the type of data that a variable can store such as integer, floating, character, etc.



There are the following data types in C language.

Types	Data Types
Basic Data Type	int, char, float, double
Derived Data Type	array, pointer, structure, union
Enumeration Data Type	enum
Void Data Type	void

BASIC DATA TYPES:

Data Types	Memory Size	Range
char	1 byte	-128 to 127
signed char	1 byte	-128 to 127
unsigned char	1 byte	0 to 255
short	2 byte	-32,768 to 32,767
signed short	2 byte	-32,768 to 32,767

unsigned short	2 byte	0 to 65,535
int	2 byte	-32,768 to 32,767
signed int	2 byte	-32,768 to 32,767
unsigned int	2 byte	0 to 65,535
short int	2 byte	-32,768 to 32,767
signed short int	2 byte	-32,768 to 32,767
unsigned short int	2 byte	0 to 65,535
long int	4 byte	-2,147,483,648 To 2,147,483,647
signed long int	4 byte	-2,147,483,648 To 2,147,483,647
unsigned long int	4 byte	0 to 4,294,967,295
float	4 byte	
double	8 byte	
long double	10 byte	

Branching and Looping

A branch is an instruction in a computer program that can cause a computer to begin executing a different instruction sequence and thus deviate from its default behavior of executing instructions in order. Common branching statements include `break`, `continue`, `return`, and `goto`.

Break

The `break` is used in one of two ways; with a `switch` to make it act like a case structure or as part of a looping process to break out of the loop. The following gives the appearance that the loop will execute 8 times, but the `break` statement causes it to stop during the fifth iteration.

Example:

```
counter = 0;
```

```
While counter < 8
```

```
    Output counter
```

```
    If counter == 4
```

```
        break
```

```
    counter += 1
```

Continue

The following gives the appearance that the loop will print to the monitor 8 times, but the continue statement causes it not to print number 4.

```
For counter = 0, counter < 8, counter += 1
```

```
    If counter == 4
```

```
        continue
```

```
    Output counter
```

```
return
```

The return statement exits a function and returns to the statement where the function was called.

Example:

```
Function DoSomething  
    statements
```

```
Return <optional return value>
```

goto

The goto structure is typically not accepted in good structured programming. However, some programming languages allow you to create a label with an identifier name followed by a colon. You use the command word `goto` followed by the label.

Example

```
goto label;           // jumps to the label
```

```
some lines of code;
```

```
some lines of code;
```

```
some lines of code;
```

```
label: some statement; // Declared label
```

```
some lines of code;
```

exit

Although `exit` is technically a pre-defined function, it is covered here because of its common usage in programming. A good example is the opening a file and then testing to see if the file was actually opened.

If not, we have an error that usually indicates that we want to prematurely stop the execution of the program. The `exit` function terminates the running of the program and in the process returns an integer value back to the operating system. It fits the definition of branching which is to jump to some other place in the program.

Arrays

An array is a sequenced collection of elements of the same data type with a single identifier name. Python lists are similar to arrays in other languages but are not restricted to a single data type. The term 'array' as used in this chapter will generally also apply to Python lists unless otherwise noted.

Arrays can have multiple axes (more than one axis). Each axis is a dimension. Thus a single-dimension array is also known as a list.

A two-dimension array is commonly known as a table (a spreadsheet like Excel is a two-dimension array). In real life, there are occasions to have data organized into multiple-dimension arrays. Consider a theater ticket with section, row, and seat (three dimensions). Most single-dimension arrays are visualized vertically.

Most programmers are familiar with a special type of array called a string. Strings are basically a single dimension array of characters.

Unlike other single dimension arrays, we usually envision a string as a horizontal stream of characters and not vertically as a list.

We refer to the individual values as members (or elements) of the array. Programming languages implement the details of arrays differently.

Because there is only one identifier name assigned to the array, we have operators that allow us to reference or access the individual members of an array.

The operator commonly associated with referencing array members is the index operator. It is important to learn how to define an array and initialize its members.

Defining an Array

Language	Example
C++	<pre>int ages[] = {49, 48, 26, 19, 16};</pre>
C#	<pre>int[] ages = {49, 48, 26, 19, 16};</pre>

Structures and Unions

Structure is a user-defined data type in C programming language that combines logically related data items of different data types together. All the structure elements are stored at contiguous memory locations. Structure type variable can store more than one data item of varying data types under one name.

Union is a user-defined data type, just like a structure. Union combines objects of different types and sizes together. The union variable allocates the memory space equal to the space to hold the largest variable of union. It allows varying types of objects to share the same location.

Syntax for Structure:

```
struct [name of the structure]
{
    type member1;
    type member2;
    type member3;
};
```

Example:

```
#include <stdio.h>
struct student {
    char name[60];
    int roll_no;
    float marks;
} sdt;
int main() {
```



```

printf("Enter the following information:\n");
printf("Enter student name: ");
fgets(sdt.name, sizeof(sdt.name), stdin);
printf("Enter student roll number: ");
scanf("%d", & sdt. roll_no);
printf("Enter students marks: ");
scanf("%f", & sdt.marks);
printf("The information you have entered is: \n");
printf("Student name: ");
printf("%s", sdt.name);
printf("Student roll number: %d\n", sdt. roll_no);
printf("Student marks: %.1f\n", sdt.marks);
return 0;
}

```

Output:

Enter the following information:

Enter student name: James

Enter student roll number: 21

Enter student marks: 67

The information you have entered is:

Student name: John

Student roll number: 21

Student marks: 67.0

Syntax for Union

```

union [name of union]
{
    type member1;
    type member2;
    type member3;
};

```

Example:

```
#include <stdio.h>
```

```

union item
{
    int x;
    float y;
    char ch;
};

int main( )
{
    union item it;
    it.x = 12;
    it.y = 20.2;
    it.ch = 'a';

    printf("%d\n", it.x);
    printf("%f\n", it.y);
    printf("%c\n", it.ch);

    return 0;
}

```

Output:

1101109601

20.199892

a

Pointers

The pointer in C language is a variable which stores the address of another variable. This variable can be of type int, char, array, function, or any other pointer. The size of the pointer depends on the architecture. However, in 32-bit architecture the size of a pointer is 2 byte.

Consider the following example to define a pointer which stores the address of an integer.

1. int n = 10;
2. int* p = &n; // Variable p of type pointer is pointing to the address of the variable n of type integer.

Example

```

#include<stdio.h>
int main(){

```

```

int number=50;
int *p;
p=&number;//stores the address of number variable
printf("Address of p variable is %x \n",p); // p contains the address of the number therefore printing p gives the address of number.
printf("Value of p variable is %d \n",*p); // As we know that * is used to dereference a pointer therefore if we print *p, we will get the value stored at the address contained by p.
return 0;
}

```

Output:

Address of number variable is fff4
Address of p variable is fff4
Value of p variable is 50

Advantage of pointer

1) Dynamic memory allocation

In c language, we can dynamically allocate memory using malloc() and calloc() functions where the pointer is used.

2) Arrays, Functions, and Structures

Pointers in c language are widely used in arrays, functions, and structures. It reduces the code and improves the performance.

Call By Value & Call By Reference

Call By Value: In this parameter passing method, values of actual parameters are copied to function’s formal parameters and the two types of parameters are stored in different memory locations. So any changes made inside functions are not reflected in actual parameters of the caller.

Call by Reference: Both the actual and formal parameters refer to the same locations, so any changes made inside the function are actually reflected in actual parameters of the caller.

Call By Value	Call By Reference
While calling a function, we pass values of variables to it. Such functions are known as “Call By Values”.	While calling a function, instead of passing the values of variables, we pass address of variables(location of variables) to the function known as “Call By References.
In this method, the value of each variable in calling function is copied into	In this method, the address of actual variables in the calling function are copied into the dummy variables of the called function.

<p>corresponding dummy variables of the called function.</p>	
<p>With this method, the changes made to the dummy variables in the called function have no effect on the values of actual variables in the calling function.</p>	<p>With this method, using addresses we would have an access to the actual variables and hence we would be able to manipulate them.</p>
<pre>// C program to illustrate // call by value #include <stdio.h> // Function Prototype void swapx(int x, int y); // Main function int main() { int a = 10, b = 20; // Pass by Values swapx(a, b); printf("a=%d b=%d\n", a, b); return 0; } // Swap functions that swaps // two values void swapx(int x, int y) {</pre>	<pre>// C program to illustrate // Call by Reference #include <stdio.h> // Function Prototype void swapx(int*, int*); // Main function int main() { int a = 10, b = 20; // Pass reference swapx(&a, &b); printf("a=%d b=%d\n", a, b); return 0; } // Function to swap two variables // by references void swapx(int* x, int* y) {</pre>

<pre>int t; t = x; x = y; y = t; printf("x=%d y=%d\n", x, y); }</pre> <p>Output: x=20 y=10 a=10 b=20</p>	<pre>int t; t = *x; *x = *y; *y = t; printf("x=%d y=%d\n", *x, *y); }</pre> <p>Output: x=20 y=10 a=20 b=10</p>
<p>Thus actual values of a and b remain unchanged even after exchanging the values of x and y.</p>	<p>Thus actual values of a and b get changed after exchanging values of x and y.</p>
<p>In call-by-values, we cannot alter the values of actual variables through function calls.</p>	<p>In call by reference we can alter the values of variables through function calls.</p>
<p>Values of variables are passed by the Simple technique.</p>	<p>Pointer variables are necessary to define to store the address values of variables.</p>

C++

OVERVIEW OF C++

- C++ is a statically typed, compiled, general-purpose, case-sensitive, free-form programming language that supports procedural, object-oriented, and generic programming.
- C++ is regarded as a middle-level language, as it comprises a combination of both high-level and low-level language features.
- C++ was developed by Bjarne Stroustrup starting in 1979 at Bell Labs in Murray Hill, New Jersey, as an enhancement to the C language and originally named C with Classes but later it was renamed C++ in 1983.
- C++ is a superset of C, and that virtually any legal C program is a legal C++ program.

CLASS AND OBJECTS:

CLASS:

When you define a class, you define a blueprint for a data type. This doesn't actually define any data, but it does define what the class name means, that is, what an object of the class will consist of and what operations can be performed on such an object.

A class definition starts with the keyword `class` followed by the class name; and the class body, enclosed by a pair of curly braces. A class definition must be followed either by a semicolon or a list of declarations. For example, we defined the `Box` data type using the keyword `class` as follows –

```
class Box {
    public:
        double length; // Length of a box
        double breadth; // Breadth of a box
        double height; // Height of a box
};
```

The keyword `public` determines the access attributes of the members of the class that follows it. A public member can be accessed from outside the class anywhere within the scope of the class object. You can also specify the members of a class as `private` or `protected` which we will discuss in a sub-section.

Example

Create a class called "MyClass":

```
class MyClass          {                //          The          class
    public:            //          Access          specifier
        int myNum;    //          Attribute          (int          variable)
        string myString; //          Attribute          (string          variable)
};
```

OBJECTS:

A class provides the blueprints for objects, so basically an object is created from a class. We declare objects of a class with exactly the same sort of declaration that we declare variables of basic types. Following statements declare two objects of class `Box` –

```
Box Box1; // Declare Box1 of type Box
Box Box2; // Declare Box2 of type Box
```

Both of the objects `Box1` and `Box2` will have their own copy of data members.

Example

Create an object called "myObj" and access the attributes:

```
class MyClass { // The class
    public: // Access specifier
```

```

    int myNum;    // Attribute (int variable)
    string myString; // Attribute (string variable)
};

int main() {
    MyClass myObj; // Create an object of MyClass

    // Access attributes and set values
    myObj.myNum = 15;
    myObj.myString = "Some text";

    // Print attribute values
    cout << myObj.myNum << "\n";
    cout << myObj.myString;
    return 0;
}

```

Multiple Objects

You can create multiple objects of one class:

Example

```

// Create a Car class with some attributes
class Car {
public:
    string brand;
    string model;
    int year;
};

int main() {
    // Create an object of Car
    Car carObj1;
    carObj1.brand = "BMW";
    carObj1.model = "X5";
    carObj1.year = 1999;

    // Create another object of Car
    Car carObj2;
    carObj2.brand = "Ford";
    carObj2.model = "Mustang";
    carObj2.year = 1969;

    // Print attribute values

```

```

cout << carObj1.brand << " " << carObj1.model << " " << carObj1.year << "\n";
cout << carObj2.brand << " " << carObj2.model << " " << carObj2.year << "\n";
return 0;
}

```

CONSTRUCTOR AND DESTRUCTOR

CONSTRUCTOR:

A constructor is a special member function of a class and shares the same name as of class, which means the constructor and class have the same name.

Constructor is called by the compiler whenever the object of the class is created, it allocates the memory to the object and initializes class data members by default values or values passed by the user while creating an object.

Constructors don't have any return type because their work is to just create and initialize an object.

Syntax of Constructor:

```

class scaler {
public:
    // Constructor
    scaler() {
    // Constructor body.
    }
};

```

DESTRUCTOR:

Default constructor is also known as a zero-argument constructor, as it doesn't take any parameter. It can be defined by the user if not then the compiler creates it on his own. Default constructor always initializes data members of the class with the same value they were defined.

```

#include <iostream>

using namespace std;
class Line {
public:
    void setLength( double len );
    double getLength( void );
    Line(); // This is the constructor declaration
    ~Line(); // This is the destructor: declaration

private:

```



```

    double length;
};

// Member functions definitions including constructor
Line::Line(void) {
    cout << "Object is being created" << endl;
}
Line::~~Line(void) {
    cout << "Object is being deleted" << endl;
}
void Line::setLength( double len ) {
    length = len;
}
double Line::getLength( void ) {
    return length;
}

// Main function for the program
int main() {
    Line line;

    // set line length
    line.setLength(6.0);
    cout << "Length of line : " << line.getLength() <<endl;

    return 0;
}

```

Inheritance

In C++, it is possible to inherit attributes and methods from one class to another. We group the "inheritance concept" into two categories:

- derived class (child) - the class that inherits from another class
- base class (parent) - the class being inherited from

To inherit from a class, use the : symbol.

In the example below, the Car class (child) inherits the attributes and methods from the Vehicle class (parent):

Example

```
// Base class
class Vehicle {
public:
    string brand = "Ford";
    void honk() {
        cout << "Tuut, tuut! \n" ;
    }
};

// Derived class
class Car: public Vehicle {
public:
    string model = "Mustang";
};

int main() {
    Car myCar;
    myCar.honk();
    cout << myCar.brand + " " + myCar.model;
    return 0;
}
```

FUNCTION OVERLOADING:

With function overloading, multiple functions can have the same name with different parameters:

Example

```
int myFunction(int x)
float myFunction(float x)
double myFunction(double x, double y)
```

Consider the following example, which have two functions that add numbers of different type:

Example

```
int plusFuncInt(int x, int y) {
    return x + y;
}

double plusFuncDouble(double x, double y) {
    return x + y;
}
```

```

}

int main() {
    int myNum1 = plusFuncInt(8, 5);
    double myNum2 = plusFuncDouble(4.3, 6.26);
    cout << "Int: " << myNum1 << "\n";
    cout << "Double: " << myNum2;
    return 0;
}

```

UNIT-II INTRODUCTION TO PYTHON

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- Python is Interpreted – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

SIMPLE PROGRAMMING

1.)

Example:

```
print ('Hello Python')
```

Output:

```
Hello World
```

2.)

Given two numbers num1 and num2. The task is to write a Python program to find the addition of these two numbers.

Examples:

Input: num1 = 5, num2 = 3

Output: 8

Input: num1 = 13, num2 = 6

Output: 19

```
num1 = 15
```

```
num2 = 12
```

```
# Adding two nos
```

```
sum = num1 + num2
```

```
# printing values
```

```
print("Sum of {0} and {1} is {2}" .format(num1, num2, sum))
```

3.)

Python Program to Check if a Number is a Palindrome

```
n=int(input("Enter number:"))
temp=n
rev=0
while(n>0):
    dig=n%10
    rev=rev*10+dig
    n=n//10
if(temp==rev):
    print("The number is a palindrome!")
else:
    print("The number isn't a palindrome!")
```

4.)

Python Program to Reverse a Number:

```
n=int(input("Enter number: "))
rev=0
while(n>0):
    dig=n%10
    rev=rev*10+dig
    n=n//10
print("Reverse of the number:",rev)
```

5.)

Python Program to Print All Odd Numbers in a Range

```
lower=int(input("Enter the lower limit for the range:"))
upper=int(input("Enter the upper limit for the range:"))
```

```
for i in range(lower,upper+1):  
    if(i%2!=0):  
        print(i)
```

UNIT-3

INTRODUCTION

Mobile Technologies

Mobile technology is a type of technology in which a user utilizes a mobile phone to perform communications-related tasks, such as communicating with friends, relatives, and others. It is used to send data from one system to another.

Portable two-way communications systems, computing devices, and accompanying networking equipment make up mobile technology.

Mobile technology is largely employed in cellular communication systems and other related areas.

It employs a network architecture that allows multiple transmitters to deliver data on a single channel at the same time. Because it reduces the potential of frequency interference from two or more sources, this platform allows multiple users to use single frequencies. The channel has evolved over time.

This is fast expanding; its applications are getting increasingly broad over time, and it is gradually replacing other similar sources of communication on the market, such as post offices and landlines.

Mobile technology has progressed from a simple phone and texting device to a multi-tasking system that can be used for GPS navigation, internet browsing, gaming, and instant messaging, among other things. With the rise, experts claim that the future of computer technology is dependent on wireless networking and mobile computing.

Through tablets and small PCs, mobile technology is becoming increasingly popular. This smartphone system has since been improved to a big multitasking computer that can be used for GPS navigation, gaming, internet browsing, and instant messaging. Tablets and portable laptops have increased the adoption of mobile technology. The mobile networks that connect these devices are referred to as wireless systems. They allow speech, data, and (mobile) apps to be shared between mobile devices.

Mobile technology is becoming increasingly prevalent. Smartphone users have surpassed 3 billion, and the global mobile workforce is expected to reach 1.87 billion by 2022. Any gadget with internet capabilities that can be accessed from anywhere is referred to as mobile technology. Smartphones, tablets, some iPods, and laptops already fall within this category, but this list will undoubtedly grow in the future years.

Types of Mobile Technologies

Followings are the few famous mobile technologies:

- SMS
- MMS
- 4G
- 3G
- GSM
- CDMA
- Wi-Fi

Let discuss them one by one in detail:

1. **SMS:** “SMS” stands for “Short Message Service.” It is now the most widely used and oldest text messaging service. SMS are also sent over cellular networks, therefore you’ll need a wireless plan and a wireless carrier.
2. SMS is fast gaining popularity in the world as a low-cost messaging medium. Every text message delivered to a cell phone has become known as SMS. Messages can usually be up to 140 characters long. SMS was originally developed for GSM phones, although it is now supported by all major cellular phone networks.

Although SMS is most commonly used for text messaging between friends or coworkers, it also has a variety of additional uses.

For example, SMS subscription services can send weather, news, sports updates, and financial quotes to consumers’ phones. Employees may also be notified of sales requests, service stops, and other business-related information via SMS.

Fortunately, text messages sent via SMS do not require the receiver’s phone to be turned on in order for the message to be delivered. The message will be kept in the SMS service until the receiver switches on his or her phone, at which point it will be transmitted to the recipient’s phone. Most cell phone providers enable you to send a specific amount of text messages per month for free.

2. MMS: MMS (Multimedia Messaging Service) messaging is a standard method of delivering multimedia material, including messages. MMS, as opposed to SMS, can send up to forty seconds of video, one picture, a multi-image slideshow, or audio. MMS texting will be supported by the majority of contemporary devices. MMS capability is typically embedded within the text message interface and is turned on automatically when needed. If you enter in a text-only message, for example, it will be transmitted by SMS. If you include a graphic or video, the multimedia part will be sent via MMS. Similarly, if someone sends you a multimedia message, your phone will automatically receive the file via MMS.

An MMS message can convey rich media content to mobile devices at any time and from any location. It is a powerful and effective tool that assists businesses in reinforcing and deepening client loyalty by providing crucial information about their products and services. Because MMS texts are packed with photographs and videos, they are a significant marketing

communication tool. As well as other audios. MMS is a cutting-edge method of communicating with others via mobile devices. Text messages are more successful because they deliver valuable information and services to the recipient. The more a corporation approaches its customers, the more probable it is to form a long-term brand partnership.

3. 3G: The third letter in the designation 3G stands for third-generation access technology, which allows mobile phones to connect to the internet. Every new technology introduces new frequency bands and data transmission rates.

The first generation emerged in the 1980s. First-generation uses large phones that had to be mounted on top of cars because they were too heavy to hold. Text messaging was made possible by the second-generation network, which became available in the 1990s. This huge and game-changing advancement also provided a more secure network and laid the path for today's ubiquitous 3G and 4G technology.

The development of 3G connection-based networks in 2001 marked the start of mainstream Internet use on mobile phones. Soon after, smartphones were introduced, bringing all of the capabilities of a device into the palm of your hand. The signals are transmitted by a network of telephone towers, ensuring robust and relatively rapid long-distance communication. The user's mobile phone is receiving data from the tower nearest to it. Although it may not appear complicated, 3G technology was revolutionary at the time it was introduced.

Upload speeds of up to 3 Mbps are possible on 3G networks. For example, about 15 seconds for uploading a 3-minute MP3 song. The fastest 2G phones, on the other hand, may get up to 144Kbps. For example, about 8 minutes to download a 3-minute song. 3G systems are intended for digital phones with a full-screen display and better connectivity.

4. 4G: The fourth generation of mobile networking technology is known as 4G, which comes after the 2G and 3G networks. Although it's commonly referred to as 4G LTE, this isn't exactly right because LTE is just one sort of 4G. Most mobile network service providers use it now since it is the most developed technology.

However, as you may have heard, 5G is becoming operational alongside current 3G and 4G mobile networks. When it initially came out, 4G revolutionized how we use the mobile internet. Despite the fact that 3G networks were relatively limited, 4G network connectivity allowed consumers to browse the internet and watch HD films on their mobile devices, thereby turning smartphones into laptops.

Most tasks that you can do on a laptop or desktop computer can now be done on mobile devices such as smartphones or tablets. No matter how much data you require, 4G networks allow you to keep consistent speeds practically anywhere. 4G was launched in the United Kingdom in 2012. Currently, the number of mobile subscribers using 3G outnumbers those using 4G. Expect this to alter in the coming years as 4G contracts become more affordable and 4G network coverage increases across the UK.

Premium 4G offers download speeds of around 14 Mbps, which is over five times quicker than the 3G network's predecessor. 4G networks can currently attain speeds of up to 150 Mbps, allowing users to download gigabytes of data in minutes, if not seconds, rather

than hours as with 3G networks. Uploading data is also significantly faster with 4G – normal upload speeds are over 8 Mbps, with theoretical rates of up to 50 Mbps, whereas 3G upload speeds are under 0.5 Mbps.

5. Global System for Mobile technology: The (GSM) is an acronym for Global System for Mobile Communication. GSM is a cellular technology that is open and digital and is used for mobile communication. It operates on the 850 MHz, 900 MHz, 1800 MHz, and 1900 MHz frequency ranges. It employs a hybrid of FDMA and TDMA.

6. Code Division Multiple Access: (CDMA) is an acronym for code division multiple access. It is a channel access mechanism that also serves as an example of multiple access. Multiple access simply means that data from multiple transmitters can be delivered onto a single communication channel at the same time.

7. Wi-Fi (Wireless Fidelity): Wi-Fi is a wireless networking technology that allows us to connect to a network or to other computers or mobile devices across a wireless channel. Data is delivered in a circular region over radio frequencies in Wi-Fi. Wi-Fi (Wireless Fidelity) is a generic acronym for a communication standard for a wireless network that functions as a Local Area Network without the use of cables or other types of cabling.

Use of Mobile technology

- The incorporation of mobile technology into business has aided telecollaboration. Now, people could connect from anywhere using mobile technology, and access the papers and documents they need to complete collaborative work.
- Work is being redefined by mobile technologies. Employees are no longer confined to their desks; they can work from anywhere in the world.
- Mobile technology can help your company save time and money. Employees who work from home save thousands on a regular basis. Mobile phones eliminate the need for costly technology like landline carrier services. Cloud-based services are less expensive than traditional systems. Technology can also help your company become more flexible and productive.
- Mobile technology has the potential to boost productivity significantly. Mobile application integration saves an average of 7.5 hours per week per employee. Workers can also become more productive with the use of smartphones and mobile gadgets.
- The popularity of cloud-based services has skyrocketed in recent years. Cloud-based mobile technology applications have been seen to be more useful than any smartphone, particularly in terms of available storage space.

Advantages of Mobile technology

- Through a variety of applications, we can now stay in touch with our friends and family members anytime we choose. We may now communicate or video visit with anybody we want by just using our cell phone or cell phone. Aside from that, the portable keeps us informed about the rest of the globe.

- Today's mobile phones have made our day-to-day activities much more natural. Today, one may check the current traffic situation on their phone and make appropriate decisions to arrive on time. The weather is also a factor.
- With the advancement of mobile technology, the entire gaming world is now under one roof. When we are tired of monotonous work or during breaks, we can listen to music, view movies, watch our favorite shows, or simply watch a video of our favorite song.
- Mobile phones are being used for a variety of legitimate tasks, including meeting schedules, sending and receiving documents, providing introductions, warnings, and job applications, among others. Cell phones have become an indispensable tool for all working people.
- These days, mobile phones are also used as a wallet to make payments. Utilities might be used to send money to friends, relatives, and others right now.

Disadvantages of Mobile technology

- The modern family has become reliant on mobile phones. In any case, when we don't have to travel, we surf the internet, play around, and create a genuine junkie.
- Because of the widespread use of mobile technology, people nowadays don't meet in person but rather tweet or comment on social media sites.
- Because of the widespread use of mobile devices, there is a major risk of losing one's protection. By efficiently reading through your web-based social networking account, anyone may now easily obtain data such as where you reside, your loved ones, what you do for a living, where you live, and so on.
- Mobile phone prices have risen in tandem with their worth. People nowadays spend a significant amount of money on cell phones, which could be better spent on more useful things like education or other beneficial items throughout our lives.

CLOUD PLATFORM:

Cloud computing applications develops by leveraging platforms and frameworks. Various types of services are provided from the bare metal infrastructure to customize-able applications serving specific purposes.

Amazon Web Services (AWS) – AWS provides different wide-ranging clouds IaaS services, which ranges from virtual compute, storage, and networking to complete computing stacks. AWS is well known for its storage and compute on demand services, named as Elastic Compute Cloud (EC2) and Simple Storage Service (S3).

EC2 offers customizable virtual hardware to the end user which can be utilize as the base infrastructure for deploying computing systems on the cloud. It is likely to choose from a large variety of virtual hardware configurations including GPU and cluster instances. Either the AWS console, which is a wide-ranged Web portal for retrieving AWS services, or the

web services API available for several programming language is used to deploy the EC2 instances.

EC2 also offers the capability of saving an explicit running instance as image, thus allowing users to create their own templates for deploying system. S3 stores these templates and delivers persistent storage on demand. S3 is well ordered into buckets which contains objects that are stored in binary form and can be grow with attributes. End users can store objects of any size, from basic file to full disk images and have them retrieval from anywhere. In addition,

EC2 and S3, a wide range of services can be leveraged to build virtual computing system including: networking support, caching system, DNS, database support, and others.

Google

AppEngine

–

Google AppEngine is a scalable runtime environment frequently dedicated to executing web applications. These utilize benefits of the large computing infrastructure of Google to dynamically scale as per the demand. AppEngine offers both a secure execution environment and a collection of which simplifies the development if scalable and high-performance Web applications.

These services include: in-memory caching, scalable data store, job queues, messaging, and corn tasks. Developers and Engineers can build and test applications on their own systems by using the AppEngine SDK, which replicates the production runtime environment, and helps test and profile applications. On completion of development, Developers can easily move their applications to AppEngine, set quotas to containing the cost generated, and make it available to the world. Currently, the supported programming languages are Python, Java, and Go.

Microsoft

Azure –

Microsoft Azure is a Cloud operating system and a platform in which user can develop the applications in the cloud. Generally, a scalable runtime environment for web applications and distributed applications is provided. Application in Azure are organized around the fact of roles, which identify a distribution unit for applications and express the application's logic. Azure provides a set of additional services that complement application execution such as support for storage, networking, caching, content delivery, and others.

Hadoop –

Apache Hadoop is an open source framework that is appropriate for processing large data sets on commodity hardware. Hadoop is an implementation of MapReduce, an application programming model which is developed by Google. This model provides two fundamental operations for data processing: map and reduce. Yahoo! Is the sponsor of the Apache Hadoop project, and has put considerable effort in transforming the project to an enterprise-ready cloud computing platform for data processing. Hadoop is an integral part of the Yahoo! Cloud infrastructure and it supports many business processes of the corporates. Currently, Yahoo! Manges the world's largest Hadoop cluster, which is also available to academic institutions.

1. Artificial Intelligence and Machine Learning
2. Data Science

3. Full Stack Development
4. Robotic Process Automation
5. Edge Computing
6. Virtual Reality and Augmented Reality
7. Blockchain
8. 5G
9. Cyber Security

1. Artificial Intelligence and Machine Learning

Artificial intelligence and machine learning once represented the cutting edge of computer science. When these technologies were created in the last 20th century, they hardly had any applications and were, in fact, mostly academic. However, these technologies have gained applications over the years and reached ordinary people's hands through their mobile phones and is currently one of the latest technologies in computer science.

Machine learning represents a computer science field in which an algorithm can predict future data based on previously generated data. Artificial intelligence represents the next step in machine learning, in which an algorithm develops data-based intelligence and can even carry out essential tasks on its own.

2. Data Science

Recent technology in computer science is Data Science. For much of the initial part of the 21st century, data science was the next big thing. Data science has been around for much longer than just the past twenty months. For centuries, data analysis has been an essential task for companies, government, institutions, and departments. Analysing data helps understand the efficiency of processes, conduct surveys of the workforce, and gauge people's general mood. However, as of today, much of data analysis has turned digital. Data analysis is among the first jobs that computers are turned to for. In the early 2000s, data analysis was so prevalent that students were being taught introductory courses on the subject in school.

3. Full Stack Development

Full-stack development refers to the development of both client-side and server-side software and is bound to be one of the top trending technologies of 2023.

The 21st century started with the dot com boom, and the internet, a relatively new phenomenon, was spreading across homes in the world. In those days, websites were no more than simple web pages, and web development wasn't the complex field that it is now.

These days, web development involves a front end and a back end. Especially for fields related to services, such as retail and e-commerce, websites include a client-side—the website that you see—and a server-side—the website that the company controls.

Generally, web developers are given the job of handling either the client-side or the website's server-side. However, being a full stack developer gives you and your company the flexibility of working on both ends of the web development spectrum. The client-side or front end will generally require knowledge of suites such as HTML, CSS, and Bootstrap. The server side requires knowledge of PHP, ASP, and C++.

4. Robotic Process Automation

Robotic Process Automation isn't just about robots. It is a lot more about the automation of processes than anything else. Before computers, most processes involved some human

intervention. Humans ran even manufacturing machines, and large-scale manufacturing employs thousands of people.

However, since computers have taken over most processes, manufacturing hasn't been left untouched either. All domains, be it manufacturing or information technology, now involve some automation in their processes. The amount of human intervention in these processes is only reducing, and this trend is likely to continue for the foreseeable future.

Jobs in robotic process automation typically involve a significant amount of coding knowledge. You would typically need to write code that would enable computerised or non-computerised processes to be done automatically without human intervention.

These processes could mean anything from automatic email replies to automated data analysis and automatic processing and financial transactions approval. Robotic process automation makes tasks considerably faster for the common consumer by making such approvals automatic based on certain conditions entered by the programmer.

In sectors such as financial services, robotic process automation can reduce the lean time to approve financial transactions online. It improves the productivity of the company as a whole, as well as that of its clients.

5. Edge Computing

During the early part of the 21st century, cloud computing was considered the next big thing. In cloud computing, data is uploaded to a centralised repository that may access it regardless of location. Cloud Computing began to be used in commercial devices only close to 2010. By the time it was 2023, cloud computing had become a prevalent technology.

In just about a decade, cloud computing had turned from being an esoteric term to being a part of a few devices in almost everybody's house. In 2023, cloud computing is no longer among the top technology trends but rather a thing of the past.

The next step after cloud computing is edge computing. It is another rising new technology in 2023 which is very similar to cloud computing, except that data is not stored in a centralised repository. In areas where network access might be difficult or impossible, cloud computing is challenging since you can no longer access the repository where your data is stored. What edge computing does is transfer data closer to the location where it needs to be used.

Edge computing has excellent applications in the Internet of Things devices. As far as IoT is concerned, a physical device you need to control with your smartphone should not need to access data from a centralised repository that might be thousands of kilometres away. Instead, data should stay as close to the device as possible.

Edge computing allows the data to remain at the 'edge' of the cloud and the device for processing so that commands can be followed through in a smaller amount of time.

Edge computing jobs have only begun to grow with IoT devices' proliferation over the past few years. As the number of these devices increases, edge computing roles are likely to become more prevalent and lucrative, placing it firmly among the trending technologies in 2021-22.

6. Virtual Reality and Augmented Reality

Virtual Reality and Augmented Reality have both been technology buzzwords for over a decade now. However, these latest technology trends in computer science have so far failed to materialise into widely available consumer products. The presence of virtual reality and

augmented reality in our real lives is minimal. Even though VR and AR have been familiar in the industry, they are relatively new technologies in 2023.

Virtual reality has been used widely in video games thus far and augmented reality-based apps did become popular for a while a few years ago, before waning. However, the best way virtual reality can become a top technology trend for the future is by making it a part of people's daily lives.

Over the past few years, virtual reality has also begun to find applications in training programs. Another domain where virtual reality experiences have been useful is in providing experiences to museum-goers. The trajectory of the rise of virtual reality is very similar to that of 3D technology—it might take just one application, such as cinema in 3D, for the technology to become mainstream. According to [Payscale](#), average salary of AR Engineer is above 6 lakhs per annum, one more reason to give this new technology a try in 2023.

Virtual reality jobs do not currently require a lot of training. Simple programming skills should be enough to land you a job, alongside an interest in the field and the power of visualisation. With millions of virtual reality devices being sold worldwide every year, it is only a matter of time before we see VR and AR take over our daily lives.

7. Blockchain

You have probably heard of Blockchain in the past few years, mostly in the context of cryptocurrency. However, Blockchain has grown to have several different applications. The significant part about Blockchain is that it is never under the complete control of a single entity due to being entirely consensus-driven. It can never change the data you store in the Blockchain used widely in sharing medical data in the healthcare industry.

Due to the security that Blockchain provides, this data can be shared among parties pretty much seamlessly. Another application of Blockchain is in maintaining the integrity of payment systems. Blockchain-based payment systems are currently highly immune to external attacks and theft. Blockchain can also be used in tracking the status of products in a supply chain in real-time.

8. 5G

If there is one technology, the knowledge of which is still little, it is 5G. It is a new technology in 2023 for which companies and governments around the world have spent years preparing for the rollout of 5G technology. In several countries, this technology has already been rolled out and achieved a significant amount of success. Since 5G is currently in a nascent stage, it is available only to a limited extent and is also relatively expensive.

The number of compatible devices with 5G is also not appreciable, although most new mobile devices being released have 5G compatibility. 5G has a much greater capacity than the current 4G technology, with an average network speed of 100 Mbps and a peak speed of 20 Gbps. If you have multiple mobile devices in your home, 5G will probably connect to these devices and use them concurrently significantly easier.

When 5G technology was only in the development stage, 5G jobs were few, and most such jobs were allocated to employees within companies. However, companies have begun to hire network engineers over the past few months, specifically for jobs associated with their 5G networks.

As 5G technology has become more prevalent, there has been a scramble among networks to purchase spectrum and roll out the technology first. This has led to the requirement of a larger workforce focussed on the development and release of 5G networks.

9. Cyber Security

The number of devices and coverage about digital technologies has been rising, hence the threat by cyber-attacks on such devices. Cyber attacks can take many forms, from phishing to identity theft, and the need to guard the greater user coverage of the internet is greater than ever.

Simple antivirus software is no longer sufficient if you want to save yourself from cyber attacks. The development of better, more sophisticated technologies to guard against cyber threats is the subject of multiple academia and industry projects worldwide.

Companies are involved not just in making new commercial technologies to protect individual domestic consumers against cyber attacks. Some of the most frequent cyber-attacks are government data repositories or the storage facilities of large companies. Nearly all large companies need a way to protect their data and their employees' data and associated firms.

Jobs in cybersecurity have been growing at thrice the pace as other tech jobs, primarily due to the reasons mentioned above. Not only are these jobs incredibly well-paying, but they are also some of the most critical positions in any firm. The 5G market in India is estimated to reach **INR 19 Billion by 2025** so this new technology in 2023 can be a game-changer.

Especially in domains such as e-commerce and retail, the importance of cybersecurity cannot be underscored. Thousands of customers store their personal and financial data on retail companies' websites to allow for easy payments. They also have accounts and passwords, which need to be protected. Similarly, in the healthcare industry, patient data needs to be protected against cyber threats.