Some Imp Ques

1. What is OOP? 2. Classes VS Objects 3. Inheritance 4. Polymorphism 5. Encapsulation 6. Abstraction 7. Operator Overloading 8. Constructor vs Destructor 9. Virtual Function 10. Methods Overriding 11. Function Overloading 12. Early Binding Vs Late Binding

OOPS: https://www.youtube.com/watch?v=m1fJjNLzRag

Q1) Real life difference between Encapsulation and Abstraction ?

Ans) Encapsulation is to hide the variables or something inside a class, preventing unauthorized parties to use. So the public methods like getter and setter access it and the other classes call these methods for accessing

Abstraction involves the facility to define objects that represent abstract "actors" that can perform work, report on and change their state, and "communicate" with other objects in the system.

Consider the below real time example:

Encapsulation: **As a driver you know how to start the car by pressing the start button and internal details of the starting operations are hidden from you. So the entire starting process is hidden from you otherwise we can tell starting operation is encapsulated from you.**

OR

The driving wheel is encapsulated the process of rotating the wheel from you.

Abstraction:

Before mentioning anything about abstraction, we can take three different users here (I am calling them as entity)

1) You 2) Local Mechanic 3) Expert

You Entity: Since you know only to start the car by pressing a button and all other operations behind the scene are abstracted from you.

Local Mechanic Entity: Our local mechanic knows some of the implementation of starting the car, i.e. he can open car's bonnet and check the battery cable or chock etc. So in short Local Mechanic Entity knows some of the implementations of the car.

Expert Entity: Since our expert (Designer of the car) mechanic knows all the operations of our car, he can repair it very quickly. So in short Expert Entity knows all the implementations of the car.

The car's operation is completely abstracted from you and it is partially implemented to Local Mechanic Entity and fully implemented to Expert Entity. So you are an abstract class having only abstract methods, Local Mechanic Entity has extended You(Since he is also an ordinary user) and he implemented some of the methods and last our expert Entity extending Local Mechanic and implementing all the methods.

Q2) Constructors

```
 Class A{
private:
        Int age;
        Public:
                A(int x){     // constructor
                        age=x;  //same name as class and don't have any return
                }
                Int getData( ){
                        Return age;
                }
};
Void main(){
        A obj(24);
        cout<<obj.getData();
}
```

// Normal class

```
Class A{
Private:
        Int age;
        Public:

                Void setData(int x){
```

```
            age=x;
        }
        Int getData(){return age;}
}
Void main(){
A obj;
obj.setData(10);    // Intermediate to first set the data and then call the getData to get it
cout<<obj.getData();
```

Q3) Types of Constructors

1) **Non parameterized constructor (default constructor)**
2) **Parameterized constructor**
3) **Copy constructor**

Non Parametrized

```
Class person{
Private:
    String name;
    Int age;
    Float height;
Public:
    person(){   -> Default constructor - just allocating space for the data variables
        name =”NULL”;
        age =0;
        height=0.0f;
    }
    Void getData(){
        cout<<name<<” “<<age<<” “<<height<<endl;
    }
};
Int main(){
    Person obj;
    obj.getData();
}
```

Parametrized - done earlier

Copy constructor

```
A(int x){
    age=x;
}
A( A &obj1){    // passing by reference so that changes get reflected
    age=obj1.age;
}
Void main(){
    A obj1, obj2;
    A obj2(obj1);
    cout<<obj2.getData();
}
```

Q4) Overloaded constructor - When we assign default value to argument of the constructor.

```
Class person{
Private:
    String name;
    Int age;
    Float height;
Public:
    person(string name1=” “, int age1=0, float height1=0.0) {   // overloaded constructor
        name =name1;;
        age =age1;
        height=height1;
    }
    person (person &obj2){   // copy constructor
        name = obj2.name;
        age = obj2.age;
        height = obj2.height;
    }
    Void getData(){
        cout<<name<<” “<<age<<” “<<height<<endl;
    }
};
Int main(){
    Person obj;
    obj.getData();
}
```

Q5) WAP, take Phone details as input ans store them in object & use Constructors

Phone details:

1) **Name**
2) **RAM**
3) **Processor**
4) **Battery**

```
Class mobile{
Private:
        String name;
        int RAM;
        string processor;
        int battery;
Public:
        mobile(string name1=”null”, int RAM1=0, string processor1=”null”, int battery1=0){
                name = name1;
                RAM = RAM1;
                processor = processor1;
                battery = battery1;
        }
        mobile(mobile &mob){
                name=mob.name;
                RAM=mob.RAM;
                processor=mob.processor;
                battery=mob.battery;
        }
        Void getMobileData();
};
// body outside the class
Void mobile:: getMobileData(){  -> scope varibles
        cout<<name<<endl;
        cout<<RAM<<endl;
        cout<<prcoessor<<endl;
        cout<<battery<<endl;
}
Int main(){
        Mobile iphone;
        Mobile nokia(“Athar”, 20, “OxygenOS”, 4500);
        Mobile samsung(nokia);

        iphone.getMobileData();
        nokia.getMobileData();
        smasung.getMobileData();
}
```

Q6) Operator Overloading

Q7) Inheritance - one class inquires the properties from other class (base class).

Grandfather <- Father <- Child
Parent Class <- Child Class
Base Class <- Derived Class

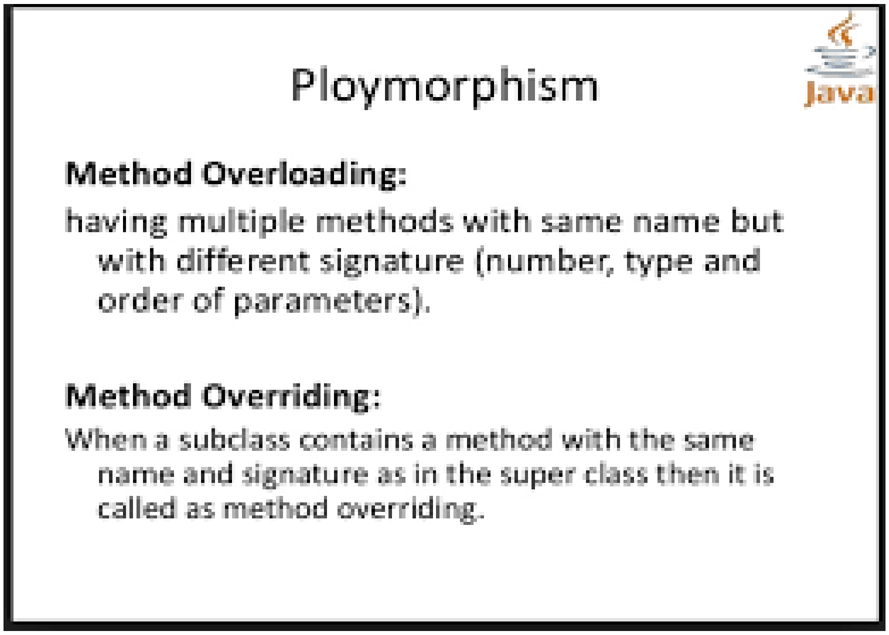
Q8) WAP in which cuboid class inherit rectangle class and calculate area and volume

```
Class rectangle{    // Base(parent class)
Public:
        int l, int b;
        Void area(){
        cout<<”area of rectangle”<<l*b<<endl;
        }
};
Class cuboid : public rectangle{   //Derived(child class)

Public:
        Int height;
        Void volume(){
                cout<<”Volume of cuboid”<<l*b*height<<endl;
        }
};
```

```
Int main(){
        cuboid c;
        c.l=10, c.b=12,c.height=3;
        c.area();
        c.volume();
}
```

Q8) Diff b/w function Overriding and function Overloading?



Ploymorphism

Method Overloading:

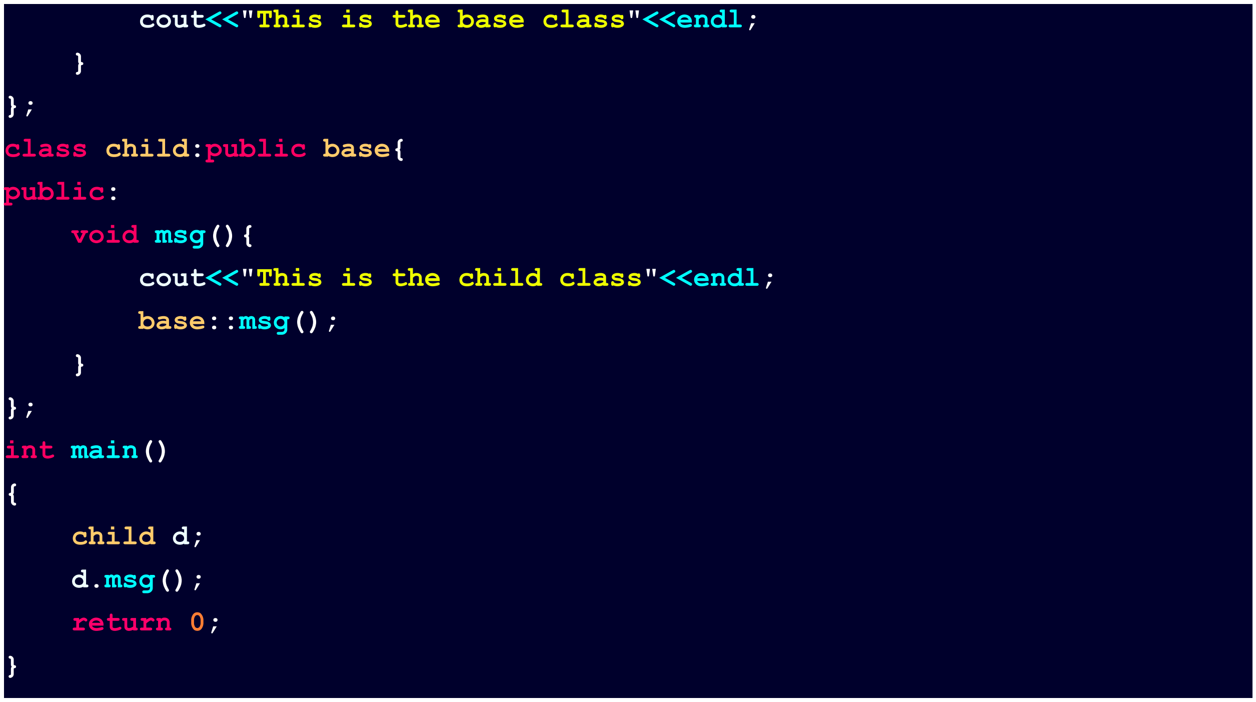
having multiple methods with same name but with different signature (number, type and order of parameters).

Method Overriding:

When a subclass contains a method with the same name and signature as in the super class then it is called as method overriding.

Overloading	Overriding
Functions have same name but different number or different type of parameters.	Functions have same name ,same number and same type of parameters.
Overloading of the functions take place at compile time.	Overriding of the functions take place at run time.
It is also known as compile time polymorphism.	It is also known as run time polymorphism.

```
// Overriding
class base{
public:
    void msg(){
```



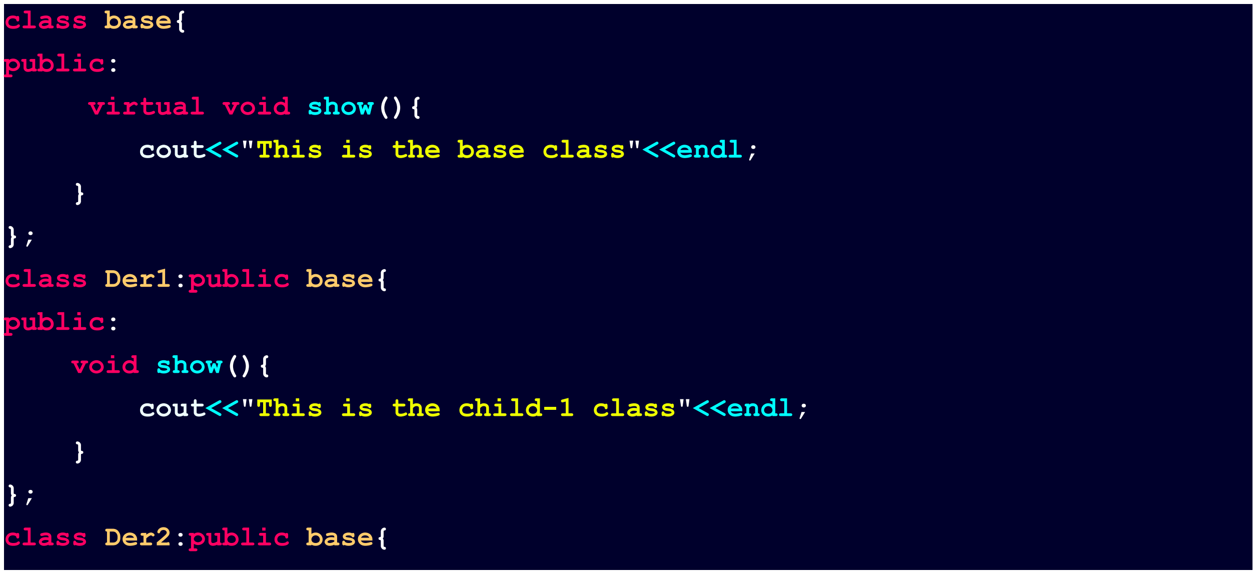
```
        cout<<"This is the base class"<<endl;
    }
};
class child:public base{
public:
    void msg(){
        cout<<"This is the child class"<<endl;
        base::msg();
    }
};
int main()
{
    child d;
    d.msg();
    return 0;
}
```

Q9) Virtual Function

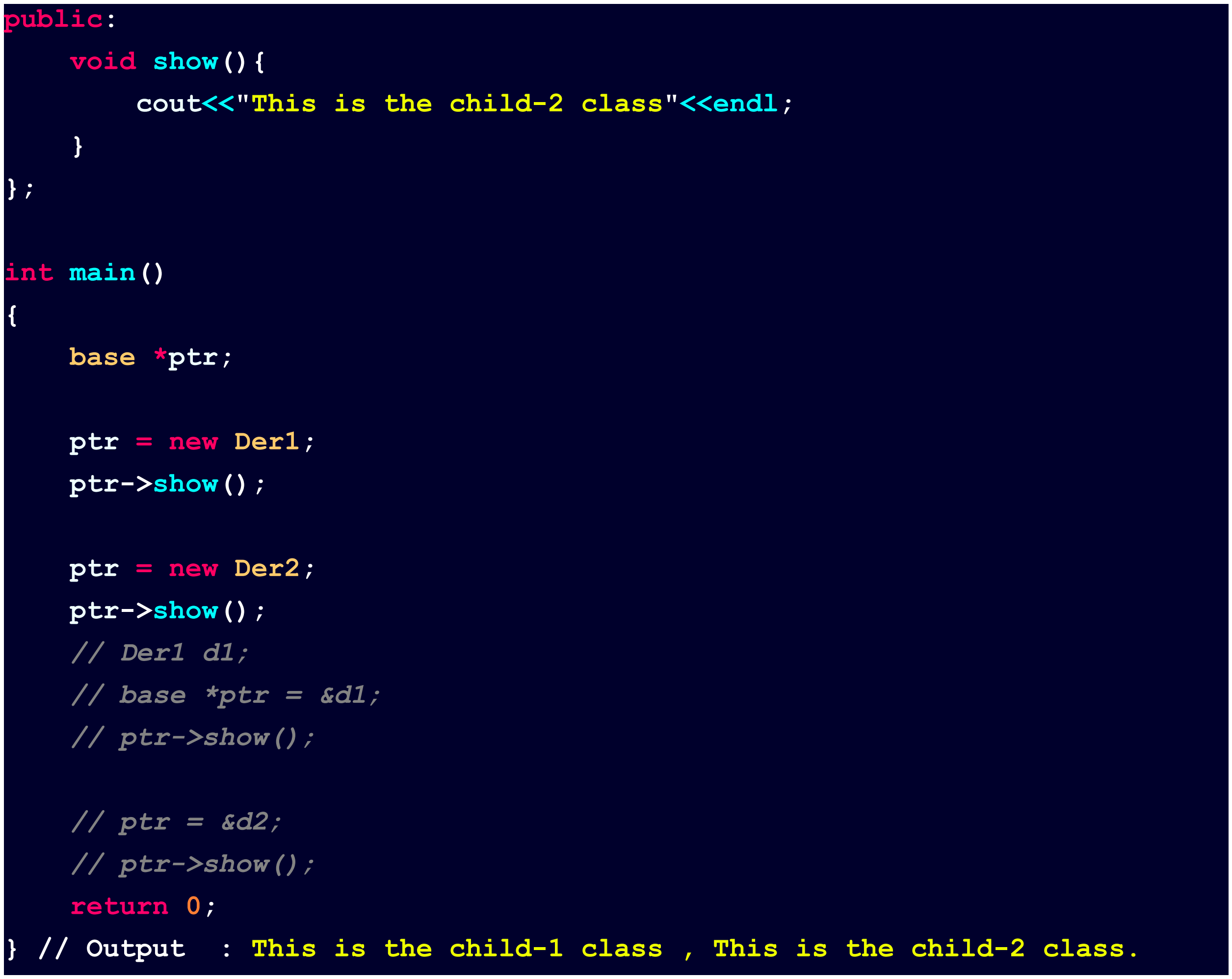
Virtual means exciting in appearance but not in reality.

Virtual function means func existing in class but can’t be used.

A virtual function (also known as virtual methods) is a member function that is declared within a base class and is re-defined (overridden) by a derived class.



```
class base{
public:
     virtual void show(){
        cout<<"This is the base class"<<endl;
    }
};
class Der1:public base{
public:
    void show(){
        cout<<"This is the child-1 class"<<endl;
    }
};
class Der2:public base{
```



```
public:
    void show(){
        cout<<"This is the child-2 class"<<endl;
    }
};

int main()
{
    base *ptr;

    ptr = new Der1;
    ptr->show();

    ptr = new Der2;
    ptr->show();
    // Der1 d1;
    // base *ptr = &d1;
    // ptr->show();

    // ptr = &d2;
    // ptr->show();
    return 0;
} // Output  : This is the child-1 class , This is the child-2 class.
```

Rules for Virtual Functions

The rules for the virtual functions in C++ are as follows:

1. Virtual functions cannot be static.
2. A virtual function can be a friend function of another class.
3. Virtual functions should be accessed using a pointer or reference of base class type to achieve runtime polymorphism.
4. The prototype of virtual functions should be the same in the base as well as the derived class.

Late Binding - Compiler differ the decision during execution of program

Q10) Abstract Class

Sometimes implementation of all functions cannot be provided in a base class because we don’t know the implementation. Such a class is called an abstract class.

Its Implementation must be provided by derived classes.

A pure virtual function (or abstract function) in C++ is a virtual function for which we can have an implementation, But we must override that function in the derived class, otherwise, the derived class will also become an abstract class. A pure virtual function is declared by assigning 0 in the declaration.

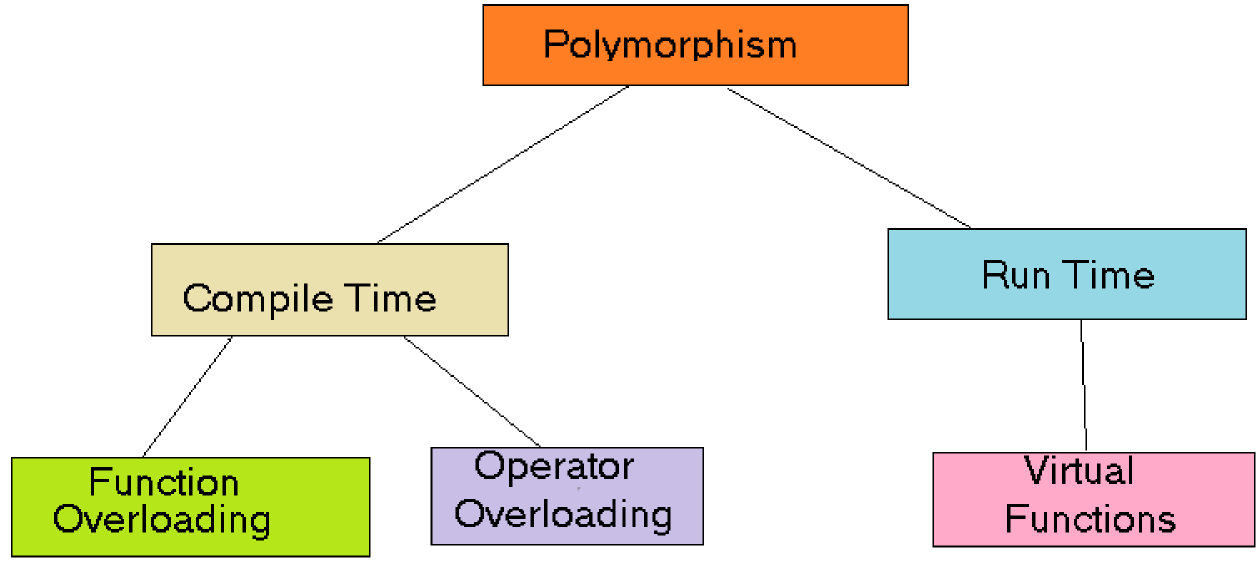
```
class parent{
public:
     virtual void show()=0; // Abstract class
    //  {
    //     cout<<"This is the base class"<<endl;
    // }
};
class boy:public parent{
public:
    void show(){
        cout<<"I am a BOY"<<endl;
    }
};
class girl:public parent{
public:
    void show(){
        cout<<"I am a GIRL"<<endl;
    }
};

int main()
{
    parent *ptr;
    ptr = new boy; // ptr store the reference of boy class
    ptr->show();


    return 0;
}
```

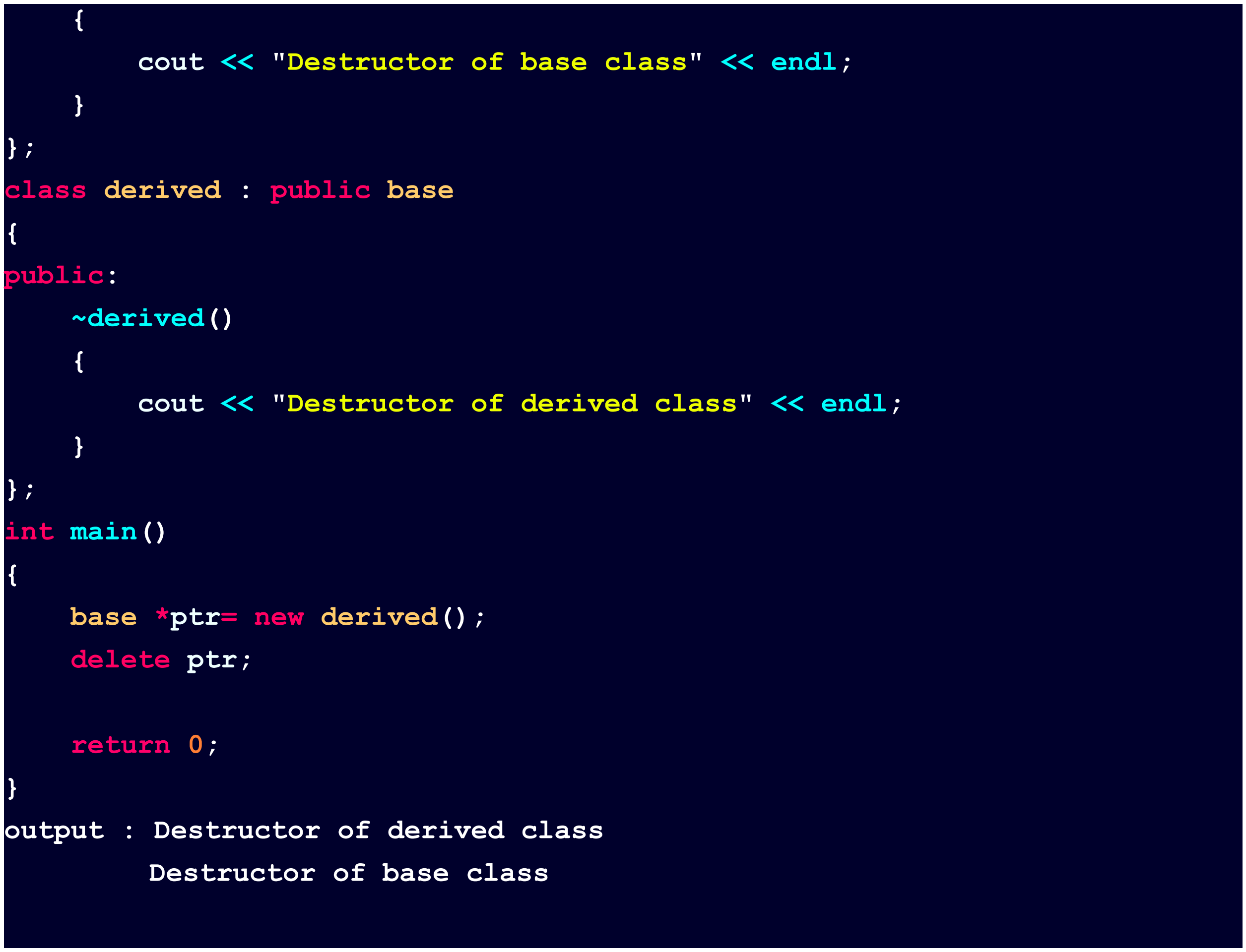
new - is used to create the object of any class.
Eg:
Base *ptr= new derived;

Q11) Polymorphism



Q12) Virtual Destructor

```
class base
{
public:
    virtual ~base()
```



```
    {
        cout << "Destructor of base class" << endl;
    }
};
class derived : public base
{
public:
    ~derived()
    {
        cout << "Destructor of derived class" << endl;
    }
};
int main()
{
    base *ptr= new derived();
    delete ptr;

    return 0;
}
output : Destructor of derived class
         Destructor of base class
```

Q13) Static Member -
It allocate memory only once.
And the memory is shared by both the objects.
Static Data members belong to a class and are common to all objects.

Q14) WAP showing ATM functionalities using OOP’s

1) **Check Balance.**
2) **Cash WithDraw**
3) **User Details**
4) **Update Mobile no.**

INTRODUCTION

Hello, Sir I am Md Athar and I am currently pursuing BTech from Electronics Engineering at IIT BHU. I have completed my 10th from ICSE board from De nobili school cmri, Dhanbad and my class 12th from CBSE board Sadar Alam memorial school, patna. I'm passionate about data structures, algorithms, and web development.

I have done an internship at Coforge as an AI/ML intern, where I applied NLP techniques to analyze customer complaints. I also developed an automated email generation system which greatly improved customer satisfaction and also designed a Streamlit web application for smooth user interaction.

In addition to my internship, I've worked on some web development projects using React, javascript and HTML/CSS. My hobbies is to play football and basketball.

INTERSHIPS

During my summer internship at Coforge, I worked on a project for a German railway company. The project focused on developing a Complaint Management System and improving customer satisfaction using advanced NLP techniques.

"My main responsibilities included analyzing 3,000+ customer complaints using NLP techniques like tokenization, stemming, lemmatization, and stopword removal. By refining the classification models, I was able to improve the system's accuracy by 15%, leading to better complaint categorization."

We need to find the sentiment score of each complaint based on that we find the positive, negative and neutral score. And for the complaint which have high negative value we need to be analyse the complaint and sent the customer the voucher code so that he could used it in next journey.

"I also designed an automated email generation system using Gemini and Llama2, which personalized responses based on the complaint type. This system not only streamlined customer communication but also improved customer satisfaction by providing quicker and more accurate responses."

"To ensure that the models and tools were accessible, I developed a user-friendly Streamlit interface. This allowed team members to interact seamlessly with the NLP models without technical expertise, significantly improving the system's usability."

"Overall, my contributions improved the classification accuracy of the complaint management system, enhanced customer satisfaction with faster email responses, and made the tools more accessible for team members. This experience honed my technical and problem-solving skills while showing me the importance of aligning technology with user needs."

Problem faced:

1) In starting we were given a total of 110 rows of dataset, so it was very difficult to create a model and check our accuracy.

2) Apply Tokenisation on different case description was bit challenging. As every complaint as different types of error like syntactic error, typing mistake, punctuation error.

3) Converting english language to vector for the machine to understand was hard part. I used one Hot encoding for that but I have to do pre-padding but then I used TF-IDF vectorizor which was more accurate and it helped convert text to vector and improved the accuracy alot.

4) Integration of LLM model was tough. First I used gpt-2 version as it was the only free model for used. But it didn't produces a better response. So my manager suggest me to use Llama2 but It was giving good response but i took a little time to generate response. Finally I have used Gemini, which was built by google It give me best response and also it was very fast.

5) Integration of model with the web application was hard so this I have used pickle which help in saving the model.

Future works:

1) sending of mail to the customer by the backend team.

2) Suppose a customer booked a ticket but 2 people got same seat so we need to give that customer a goodwill voucher. This was going to be done by the web application for the backend team of railway company.

3) Analyse different complaint and send to respective department for improving it as soon as possible.

Q) What specific NLP techniques did you use to analyze customer complaints, and how did they improve classification accuracy by 15%?

1. Text Preprocessing

Lowercasing: Converting all text to lowercase to avoid treating the same word in different cases as separate entities (e.g., "Complaint" and "complaint").
Stopword Removal: Common words like "the", "is", and "in" were removed using libraries like NLTK or SpaCy. This reduced noise in the text and focused on important words.
Punctuation Removal: Special characters and punctuation were eliminated to clean the data further.
Tokenization: Splitting the text into individual words (tokens), which helped in converting raw text into a structured form that machine learning models can process.
Impact: These preprocessing steps standardized the text and reduced irrelevant information, making it easier for the model to focus on the key terms related to the complaints. This helped increase the overall accuracy by ensuring cleaner input data.

2. Stemming and Lemmatization

To reduce words to their root form and avoid treating similar words as different entities, I used:

- **Stemming:** A more aggressive technique that chops off word endings. For example, “running” becomes “run”.
- **Lemmatization:** This method considers the context and reduces words to their base or dictionary form (e.g., "better" becomes "good").

  Impact: By using lemmatization, the vocabulary size was reduced without losing the semantic meaning of the words. This led to better generalization in the classifier, contributing to improved accuracy.

3. TF-IDF (Term Frequency-Inverse Document Frequency)-> Gives more weightage to the terms that are rarely present.(like -not)

TF-IDF was applied to convert the customer complaints into numerical vectors that captured the importance of words in the documents:

- **Term Frequency (TF):** Measures how frequently a term occurs in a document.
- **Inverse Document Frequency (IDF):** Reduces the weight of terms that are common across all documents.
  Impact: TF-IDF helped emphasize the more relevant words in the complaints while reducing the weight of common, less informative words. This allowed the model to better differentiate between various complaint categories and improved classification accuracy.

4) n-Grams Analysis

n-Grams (bigrams and trigrams) were used to capture sequences of words, which helped in identifying common phrases or recurring issues that a single word might miss.

Impact: Incorporating bigrams and trigrams allowed the model to understand word sequences and the context of the complaints better. This was especially useful for identifying phrases such as "late delivery" or "poor service", improving the classification of related complaints.

5. Sentiment Analysis

Sentiment analysis was applied to determine the emotional tone of the complaints (positive, negative, or neutral). This additional feature was integrated into the classification model.

Impact: Knowing whether a complaint carried a positive or negative sentiment helped classify the severity of complaints more accurately. Complaints with highly negative sentiment were often associated with more critical issues, which led to improved targeting of complaints to appropriate categories.

Q) What more I might have used ?

1. Word Embeddings (Word2Vec)

Word embeddings like Word2Vec or GloVe were experimented with to capture the semantic meaning of words by placing them in a high-dimensional vector space based on their contextual usage.

Impact: By using word embeddings, the model could capture the relationships between words more effectively. For example, it could understand that "late" and "delay" are semantically similar and should be treated similarly in the classification task. This helped the model generalize better and classify complaints more accurately.

2.Topic Modeling (LDA)

I also employed **Latent Dirichlet Allocation (LDA)** to uncover hidden topics within the complaints. This technique helped to cluster similar complaints and identify the major themes or recurring issues.

Impact: By identifying the underlying topics, the model could more accurately group complaints into relevant categories. This improved the classifier’s ability to distinguish between similar complaints that might otherwise be confused.

Q) How These Techniques Improved Classification Accuracy

Cleaner Data: Preprocessing (lowercasing, stopwords removal, tokenization, etc.) and lemmatization made the text more uniform, improving the classifier’s focus on important features.
Contextual Understanding: Using n-grams and word embeddings helped capture context, making it easier for the classifier to correctly interpret and classify the complaints.
Relevance and Weighting: TF-IDF and NER ensured that the most important words and phrases (like product names, specific issues) were given more weight in classification.
Capturing Sentiment: Sentiment analysis added an extra layer of information, helping classify the severity and nature of complaints more accurately.

PROJECTS

Coin Bazaar - The project is build under **hackathon** which organized under **udyam** iit bhu. It was an stock trading website and Its primary objective was to facilitates the **buying and selling of financial securities**, such as **stocks, bonds**, and mutual funds, on behalf of its clients.

We had a group of 3 people including me . We made this website using react and node js and integrated different api for fetching the current market info and as a part of team my primarily role was on front-end development and design the UI and UX of website.

The major features of the project includes
I created a stock trading web application using React library and other programming languages such as javascript, html/css, node.js.
Here the some feature of my website.

User Registration: Allow users to create accounts and authenticate themselves securely.
User Dashboard: Enable users to see the details of their stock like current portfolio, account balance, transaction history, and other relevant information.

Market: In the market section, users can access a comprehensive list of available coins for trading. Each coin will have its name, symbol, current price, price changes over time, and other relevant market data.
News and Insights: Provide access to real-time market news, analysis, and insights to help users make informed trading decisions.
Responsive design : We ensure that the website in fully responsive in different devices like desktop, tabs and mobile phones.

Buying Coins: To buy their favorite coin, users can enter the quantity of the coin they want to purchase and click on the "Buy" button. The system will calculate the total cost based on the current market price of the coin and deduct it from their account balance.

Selling Coins: Users can also sell the coins they own through the platform. They select the coin they want to sell, specify the quantity, and click on the "Sell" button. The system will credit their account with the proceeds of the sale based on the current market price.

We also helped the collage student who are Engaging in trading activities them to gain practical knowledge and get hand to experience in managing their finances.

What to learn from this project ?

1) **Management Skills**
2) **Project assignment to different team members**
3) **Leadership skills and Teamwork**
4) **In React- use of router and Routing features , fetching API's**
5) **Improving UI /UX and responsiveness of website**
6) **Presentation skills**
7) **I also motivated me to do real trading which would be helpful in future in financial management.**

Problems:

1) **Real time integration of coins data was not done on our website . I use useState for it that time . To overcome this problem I used useEffect which re-render the data continuously.**
2) **API was not working at last time as there was some limit on use of that api.**
3) **We were told to make the website completely responsive, so we also spend alot of time in designing and improving the responsiveness and performance of the website.**

I learnt the value of team work and not giving up till last moment things from this project and remaining updated is very important we got to know that the api which we were using stopped working for some
We got 3rd rank in the hackathon

Loan Prediction Model:

"The goal of the project was to develop a model that predicts the likelihood of loan approval based on applicant data. This kind of predictive system can be incredibly useful for financial institutions to streamline their loan approval process and reduce manual effort."

"To tackle this, I used a dataset containing applicant information and loan statuses. I started with **data preprocessing**, which involved handling missing values, applying **label encoding** for categorical variables, and performing **feature engineering** to extract meaningful features. These steps ensured that the data was clean and ready for modeling."

"I used an **SVM classifier** because it works well for classification problems, especially when the data is not linearly separable. After training and testing the model, it achieved an **83% accuracy on the test data**, which was a significant indicator of its reliability."

"To make the project more interactive, I integrated a **data visualization tool**. Users could upload any dataset to generate various plots like scatter, count, line, and bar charts, helping with **Exploratory Data Analysis (EDA)**. This feature was especially useful for understanding data patterns before making predictions."

"This project not only improved my understanding of **machine learning workflows** but also helped me develop skills in **data preprocessing**, **model evaluation**, and **deployment**. By the end, the model achieved **83% accuracy**, and the user-friendly Streamlit app enhanced accessibility and usability."

2.EES TECH TEAM

This was a team project which was a full stack website for electronic engineering society and and all the events,competitions coming under the society. So basically this website was the backbone for our society which helped society to organize different competitions gave student notifications about upcoming events,registering for events, giving info about competition etc

As a member of the development team, my role was primarily focused on **front-end** development and user interface design. I collaborated with the back-end to integrate the front-end functionalities and ensure a seamless user experience. I also took the lead in conducting testing to refine the system's usability

We used react js for frontend and django for the backend

This was big project it took 3 months for us to develop the site there were 14 people working on this project

Finally we successfully able to create the site which was backbone of all the events organised

3. Diagnosis Of Lung Condition

It was an machine learning project where i developed a model to detect the crackles and wheezes in patient's lungs by taking the audios samples of the patient's to perform diagnosis of the patient's lung condition—wheezes and crackles—through the use of a computer vision (machine learning) algorithm.

We are provided with the audio file and text file the audio file contains the respiration cycles of patients recording and text file contains the from which timestamp the patients respiration cycle starts and ends in the audio file and the target variables that is patient is having wheezes and crackles

1) So at the start of the project first target was to convert the audio data into numerical data for analysis so i have used librosa library for this purpose.

2) Then the second target was to used the numerical data to get the all the feature so i have used the mfcc (mel frequency cepstral coefficient) technique for extracting the feature.
3) Now we are using the feature to plot the spectrograph by using librosa.specshow function.
4) Now we are resizing all the images into same size and training our model by using CNN on the image by using the target values.

After the training and testing i got approax 80% accuracy on the given model.

Q) How this is befinial for others?
Q) How you can scale this model?
Q) What are the application of this?

Common Challenges in Preprocessing Audio Data with Librosa

1) **Noise and Audio Quality**

- **Challenge**: Lung sound recordings can have a lot of background noise (e.g., patient movement, environmental sounds, stethoscope handling noise), which may interfere with the key features (crackles, wheezes) that you're trying to detect.
- **Denoising algorithms** or **thresholding techniques** can help remove background noise while preserving important features of the lung sounds.

2) **Feature Extraction**

- **Challenge**: Deciding which features to extract and how to represent them is crucial. Standard audio features like **MFCC** (Mel-frequency cepstral coefficients), **Chroma**, and **Spectral Contrast** may not always capture subtle differences in lung sounds.

3) **Handling Class Imbalance**

- **Challenge**: In lung sound datasets, you may encounter class imbalance, with far fewer examples of crackles or wheezes compared to normal breathing.

- Solution:
- **Oversample** the minority classes using techniques like SMOTE (Synthetic Minority Over-sampling Technique).

Computational Resources

- **Challenge**: Processing a large number of audio files into spectrograms, especially if they are long or high-quality, can be computationally expensive and time-consuming.

Solution : Use **GPU acceleration** where possible to speed up feature extraction and model training.

Q) Difference between crackles and wheezles?

Crackles (Rales)

Characteristics:

- **Sound Description**: Crackles are brief, discontinuous, popping sounds that resemble the sound of crumpling cellophane, snapping, or rubbing hair between your fingers near the ear.

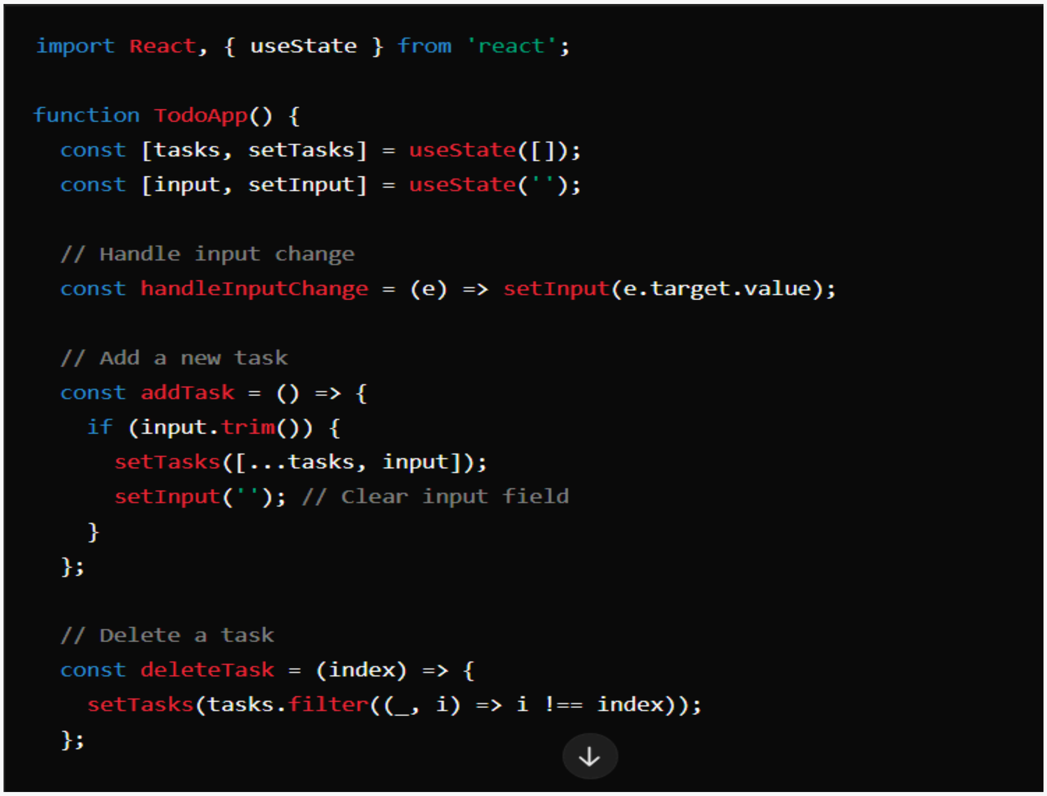
Timing: Crackles are most commonly heard during **inspiration** (when the person is inhaling), though they can sometimes be heard during expiration as well.

Wheezes

Characteristics:

- **Sound Description**: Wheezes are continuous, musical, high-pitched whistling sounds. They can vary in tone and duration but generally sound like a high-pitched whine or squeak.
- **Timing**: Wheezes are more commonly heard during **expiration** (when the person exhales), but they can occur during inspiration in severe cases.

Simple TODO APP:



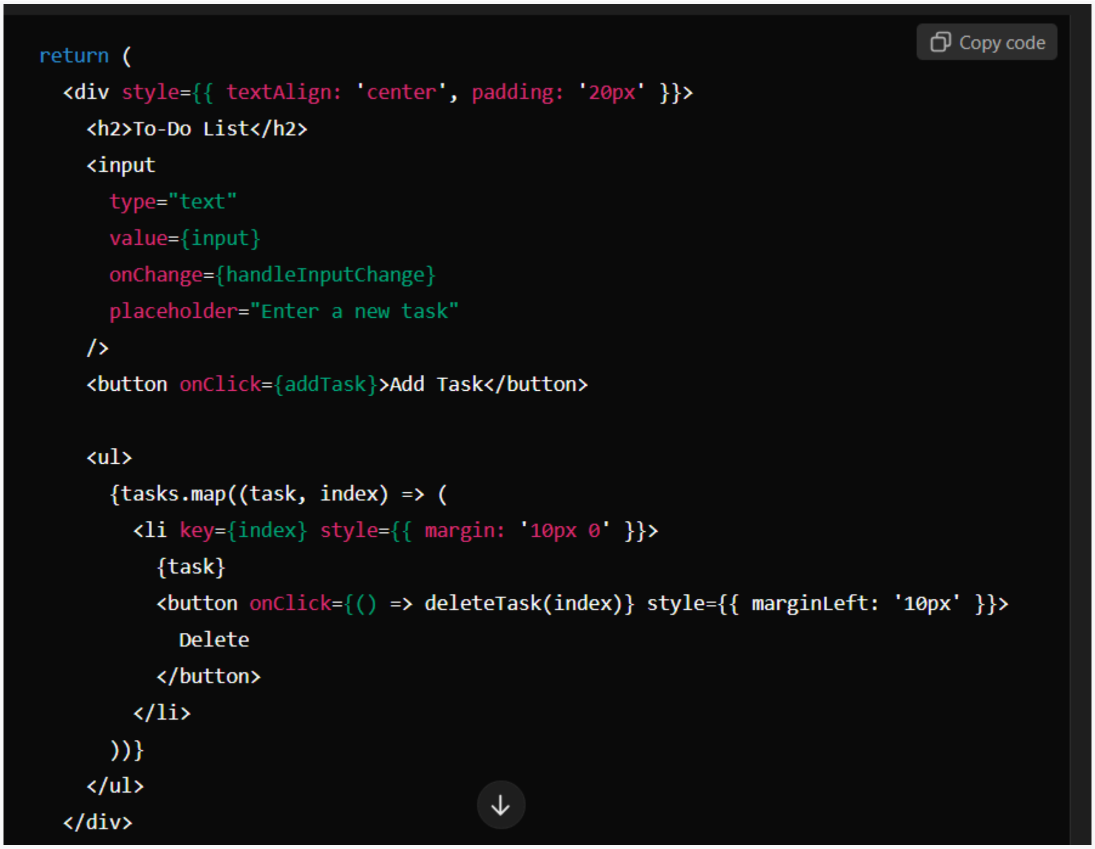
```
import React, { useState } from 'react';

function TodoApp() {
  const [tasks, setTasks] = useState([]);
  const [input, setInput] = useState('');

  // Handle input change
  const handleInputChange = (e) => setInput(e.target.value);

  // Add a new task
  const addTask = () => {
    if (input.trim()) {
      setTasks([...tasks, input]);
      setInput(''); // Clear input field
    }
  };

  // Delete a task
  const deleteTask = (index) => {
    setTasks(tasks.filter((_, i) => i !== index));
  };
```



```
return (
  <div style={{ textAlign: 'center', padding: '20px' }}>
    <h2>To-Do List</h2>
    <input
      type="text"
      value={input}
      onChange={handleInputChange}
      placeholder="Enter a new task"
    />
    <button onClick={addTask}>Add Task</button>

    <ul>
      {tasks.map((task, index) => (
        <li key={index} style={{ margin: '10px 0' }}>
          {task}
          <button onClick={() => deleteTask(index)} style={{ marginLeft: '10px' }}>
            Delete
          </button>
        </li>
      ))}
    </ul>
  </div>
```

SOME ML IMP QUES

Q1) why is CNN better than ANN?

Ans) Speed: CNNs are faster than ANNs when dealing with large amounts of data.
Accuracy: CNNs tend to be more powerful and accurate when solving classification problems.
Efficiency: CNNs are more efficient and scalable than ANNs.
One of the advantages of using CNN over ANN is that it reduces no of parameters i.e weights and bias .

Q2) What are filter in CNN?

Ans) In Convolutional Neural Networks (CNNs), a filter is a small matrix of numbers that is convolved with input data to extract certain features. Filters are also known as kernels

Q3) How to decide the number of filters to take in CNN?

Ans) There is no definite rule for choosing the number of filters in a convolutional neural network (CNN).
The more features you want to capture (and are potentially available) in an image the higher the number of filters required in a CNN.

Q4) what is keras?

Ans) **Keras** is a high-level python deep learning API used for easy implementation of Neural Networks. It has multiple low-level backends like TensorFlow, Theano, PyTorch, etc., which are used for fast computation.

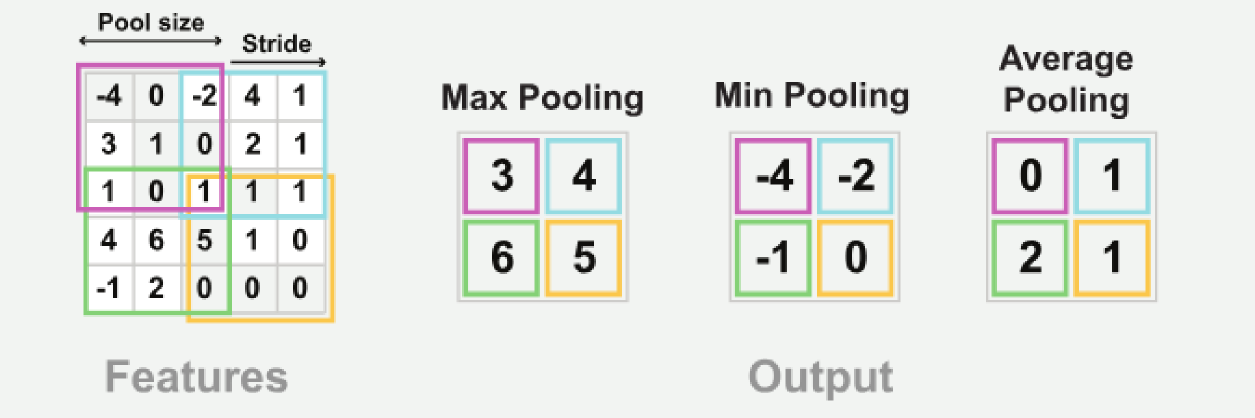
Q5) what is keras Tuner?

Ans) Keras Tuner is a hyperparameter optimization framework that makes it easy to find the best hyperparameters for your Keras models.

Q6) what is Pooling?

Ans) The main purpose of pooling is to reduce the size of feature maps, which in turn makes computation faster because the number of training parameters is reduced.

To reduce the spatial size of the input image, so that number of computations in the network are reduced.



-4	0	-2	4	1
3	1	0	2	1
1	0	1	1	1
4	6	5	1	0
-1	2	0	0	0

Max Pooling

3	4
6	5

Min Pooling

-4	-2
-1	0

Average Pooling

0	1
2	1

Q7) What is Padding?

Ans) Padding in Convolutional Neural Networks (CNNs) is the process of adding extra pixels to an input image before applying convolution. These additional pixels act as a protective boundary around the image, allowing the network to retain more spatial information.

Padding allows for more accurate analysis of images. It also helps the kernel to improve performance, especially when used to detect the borders of an image.

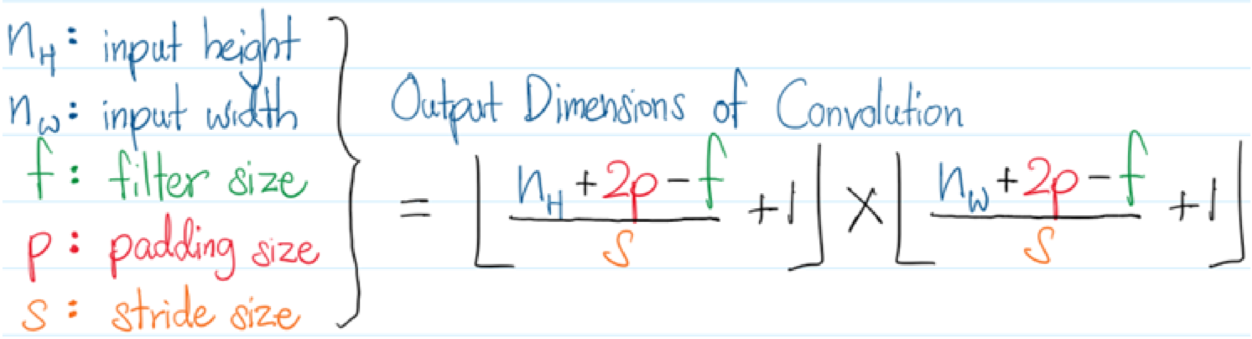
Q) Different Types of Padding?
Ans)
1) Valid (No Padding): Valid padding, also known as "No padding," means no extra pixels are added to the input data before applying the convolution operation.
(n-f+1)*(n-f+1).

2) Same (Zero Padding): These extra pixels are typically filled with zeros, which is why it's called "Zero padding." Padding element are added all around the output matrix. It will have the same dimensions as the input matrix.
(n+2p-f+1)*(n+2p-f+1) = n*n
n+2p-f+1 = n
2p=f-1
p = (f-1)/2

3) Full Padding: used in specific situations where you want the output feature map to be larger than the input feature map.

Q) What is Strides in CNN?

A) **Stride is the number of pixels shifts over the input matrix.**
When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on.



n_H : input height
n_w : input width
f : filter size
p : padding size
s : stride size

Output Dimensions of Convolution

$$= \left\lfloor \frac{n_H + 2p - f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n_W + 2p - f}{s} + 1 \right\rfloor$$

Q) Binary Cross Entropy and Category cross entropy
A) **Binary Cross Enrtopy**: It measures the dissimilarity between the predicted probability distribution and the true binary labels.
The formula for binary cross-entropy is as follows: L(y, p) = - [y * log(p) + (1 - y) * log(1 - p)] where:

- $L(y, p)$ is the loss value.
- y is the true binary label (0 or 1).
- p is the predicted probability of belonging to class 1 (i.e., the probability of being in class 1).

Category Cross Enrtopy: It measures the dissimilarity between the predicted probability distribution over multiple classes.(like cats, dogs, or birds).
The formula for categorical cross-entropy is as follows: $L(y, p) = - \Sigma(y_i * \log(p_i))$ where:

- $L(y, p)$ is the loss value.
- y_i is the true one-hot encoded label for class i (1 for the true class, 0 for others).
- p_i is the predicted probability of belonging to class i.

Q) What is Overfitting?
A) Overfitting is a common problem in machine learning and deep learning, where a model learns the training data too well, to the point that it starts to capture noise and random fluctuations in the data, rather than the underlying patterns. This results in a model that performs exceptionally well on the training data but poorly on new, unseen data.
Q) what is Underfitting?
A) Underfitting is another common problem in machine learning and deep learning, and it is the opposite of overfitting. Underfitting occurs when a model is too simple to capture the underlying patterns in the data, resulting in poor performance on both the training data and new, unseen data.

Q) Differentiate Machine Learning and Deep Learning?
A)1) **Machine Learning (ML)**: Machine learning encompasses a broad range of techniques and algorithms that enable computers to learn from data without being explicitly programmed. ML includes both traditional statistical methods (e.g., linear regression, decision trees, support vector machines) and modern techniques (e.g., random forests, gradient boosting, k-nearest neighbors).

Deep Learning (DL): Deep learning is a subset of machine learning that focuses on neural networks with many layers (deep neural networks). These deep neural networks are capable of automatically learning intricate patterns and representations from large amounts of data. Deep learning models are typically deep, complex, and consist of multiple layers, making them well-suited for tasks such as image and speech recognition, natural language processing, and more.

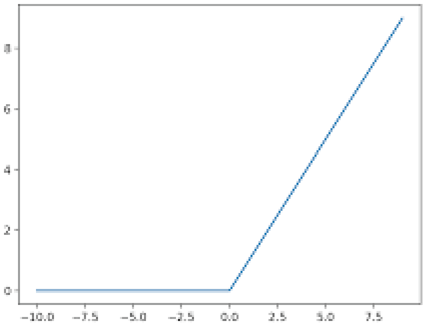
2) Hardware and Computation:

- Machine Learning (ML): Traditional machine learning models are typically less computationally intensive and can run on standard hardware.
- Deep Learning (DL): Deep learning models, especially deep neural networks, require specialized hardware like GPUs and TPUs for training and inference due to their computational complexity.

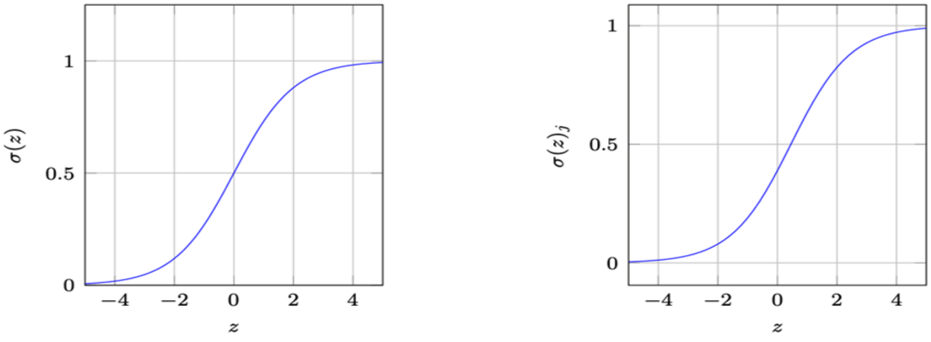
Q) Application of Deep Learning?
A) Computer Vision, Image Recognition, NLP, Object Detection.

Q) What is Activation Function ?
A) The activation function introduces non-linearity to the model, enabling the neural network to learn complex patterns and translate inputs into a usable output parameter.
Many types of activation functions:

1) ReLU: The function returns 0 if it receives any negative input, but for any positive value x it returns that value back. So it can be written as f(x)=max(0,x).



2) Softmax: The output of the softmax function is a **vector of probabilities,** one for each class, with the property that the **probabilities sum to 1.** In a **multi-class classification task**, the class with the highest probability is typically chosen as the predicted class.

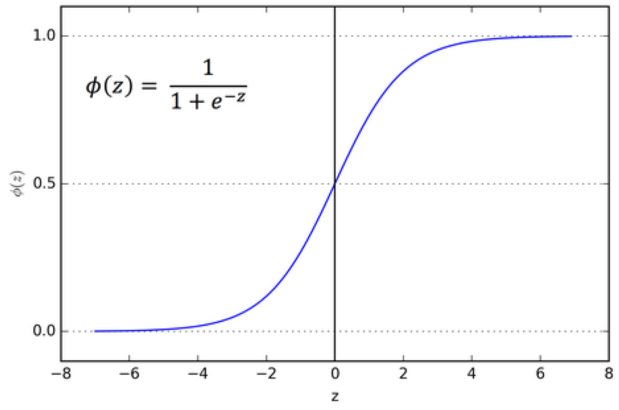


(a) Sigmoid activation function.

(b) Softmax activation function.

Figure 1: Sigmoid and Softmax activation functions

3) Sigmoid: The sigmoid function maps its input to a range between 0 and 1.



Q) Why is Fourier Transform used in Deep Learning?
A) It is an effective package used for analyzing and managing large amounts of data present in a database. It ensures the high efficiency is maintained.

Q) What are loss function?
A) Loss function is used to measure the accuracy of the model. This is done by comparing the training dataset and testing dataset.
It’s primary goal is to measure the performance of the neutral network.
Some Examples:

1) Mean Squared Error (MSE) Loss: Used for regression problems.

#) Measures the average of the squared differences between predicted and true values.
Example: Predicting house prices, where the loss measures the difference between predicted and actual prices.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE = mean squared error
n = number of data points
Y_i = observed values
$\hat{Y}_i$ = predicted values

2) Binary Cross-Entropy Loss (Log Loss): READ FROM ABOVE.
3) Categorical Cross- Entropy: READ FROM ABOVE.

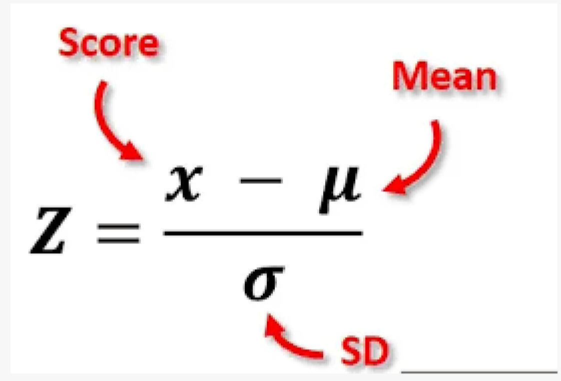
Q) Some of the framework or tools used in Deep Learning?
A) Tensorflow, Keras, PyTorch, Caffe2, MXNet etc.

Q) What is Data Normalization in Deep Learning?
A) It is a preprocessing step that is used to refit the data into a specific range.
Example:

1) Min-Max Scaling (Feature Scaling):

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

2) Z-Score Normalization:



$$Z = \frac{x - \mu}{\sigma}$$

Q) Forward propagation -> the process begins from the input layer and moves toward the final output layer.
Q) Backpropagation -> is the process begins from the output layer and moves backward to the input layers.

Q) what are Hyperparameter?
A) Hyperparameters are parameters that are not learned by a machine learning model during training but are set prior to training. They play a crucial role in controlling the behavior and performance of the model.
Examples;

1) Learning Rate: It determines the step size or rate at which the model's parameters are updated during training. eg-> gradient descent.
2) Number of Epochs:how many times the entire training dataset is passed forward and backward through the neural network during training.
3) Batch Size:
4) Number of Layers
5) Dropout Rate:The dropout rate is a hyperparameter that determines the proportion of neurons to drop out.
6) Activation Functions:
7) Optimization Algorithm: The choice of optimization algorithm, such as gradient descent, Adam, or RMSprop, is a hyperparameter that **affects the convergence and stability of training.**

Q) Dropout?
A) is a technique that is used to avoid overfitting a model in DEEP LEARNING.

Q) Tensors-> multidimensional arrays in Deep learning that are used to represent data.

Q) CNNs -> convolutional neural network are used to perform analysis on images and visuals.

Q) 3 main supervised learning methods

1) ANN(Artificial Neural Networks)
2) CNN (Convolutional neural network
3) RNN (Recurrent neural network

Q) what is Bagging?

A) is the concept of splitting a dataset and randomly placing it into bags for training the model.

Q) what is Boosting?

A) This is used to retrain the model and increase accuracy.

HR QUES

q1) Why do u want to join our company ?

Ans) I believe that working in your organization would provide me with the opportunity to combine my professional expertise with my passion for making a difference. I am eager to contribute to your company's mission and be part of a team that shares my values and is committed to creating a better future.

Q) Tell me about a challenging situation you've faced at work and how you resolved it.

A) Use the STAR method (Situation, Task, Action, Result) to structure your response. Describe the specific situation, what you had to do, the actions you took, and the positive outcome.

Q) Where do you see yourself in five years?

Ans)In five years, I see myself continuing to work hard and progressing in my current role. I hope to have acquired the necessary skills and experience to successfully lead a team where I want to be seen as a valuable member. I would also like to have developed a good working relationship with all team members.

Q) How do you handle stress and pressure?

Ans)Staying positive.

- **Using stress as a motivator.**
- **Accepting what you can't control.**
- **Practicing relaxation methods, like yoga or meditation.**
- **Choosing healthy habits.**
- **Learning how to manage time better.**
- **Making time for your personal life.**