VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

**Subject Code: CSE1002**

**Subject Name: Problem Solving and Object Oriented Programming**

# University Teaching Assistantship Report

Submitted by

**Name: Pranav Motarwar**

**Reg.No: 18BCE1015**

In partial fulfillment for the certificate of the appreciation of

**UNIVERSITY TEACHING ASSISTANTSHIP**

Under the mentorship of

**Faculty Name:  Prof. Bharathi raja**

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

WIN 2020-21

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

**School of Computer Science and Engineering**

# CERTIFICATE OF COMPLETION

This is to certify that University Teaching Assistantship for the course **CSE1002 - Problem Solving and Object Oriented Programming** has been successfully completed by **Pranav Motarwar (18BCE1015)** under the guidance of **Prof. Bharathi Raja,** Faculty-in-charge of the course.

Signature of Faculty

## SESSION LOG

| Sr. No. | Date | Hours | Topics Covered | No. of Students Attended |
|---|---|---|---|---|
| 1 | 09/04/21 | 1 | Introduction to OOPS | 33 |
| 2 | 10/04/21 | 1 | Basics about Function | 33 |
| 3 | 11/04/21 | 1 | Basics about Class | 33 |
| 4 | 16/04/21 | 1 | Basics about Class (Q) | 30 |
| 5 | 17/04/21 | 1 | Basics about Object | 30 |
| 6 | 17/04/21 | 1 | Basics about Object (Q) | 30 |
| 7 | 18/04/21 | 1 | Abstraction | 36 |
| 8 | 23/04/21 | 1 | Abstraction (Q) | 36 |
| 9 | 23/04/21 | 1 | Virtual Functions | 36 |
| 10 | 25/04/21 | 1 | Inheritance | 28 |
| 11 | 30/04/21 | 1 | Inheritance (Q) | 28 |
| 12 | 01/05/21 | 1 | Encapsulation | 28 |
| 13 | 02/05/21 | 1 | Encapsulation (Q) | 33 |
| 14 | 14/05/21 | 1 | Polymorphism | 33 |
| 15 | 15/05/21 | 1 | Polymorphism (Q) | 33 |
| 16 | 16/05/21 | 1 | Top 50 OOPS Questions | 40 |
| 17 | 21/05/21 | 1 | Placement OOPS Questions | 40 |
| 18 | 22/05/21 | 1 | Placement OOPS Questions | 40 |
| 19 | 23/05/21 | 1 | Placement OOPS Questions | 40 |

*Q - Solved questions session

Total Number of Hours: 19

## Questions Solved

Question 1: To check the survival score of the COVID19 Patient based on the input Age, CT Score, Vaccinated Status.

Initial Survival Score = 100

Condition 1:

| | |
|---|---|
| Age (0 - 18) | Score - 0 |
| Age (19 - 45) | Score - 2 |
| Age (46 - 65) | Score - 5 |
| Age (66 - 100) | Score - 10 |

Condition 2:

| | |
|---|---|
| CT Score (0 - 10) | Score - 2 |
| CT Score (11 - 20) | Score - 5 |
| CT Score (21 - 25) | Score - 10 |

Condition 3:

| | |
|---|---|
| Vaccinate status 1 | Score + 2 |
| Vaccinate status 2 | Score + 5 |

Input Format:
First Line: Vaccinated status (1: Partial; 2: Full)
Second Line:  Age (Range 0-100)
Third Line: CT score (Range 0-25)

Output Format:
First Line: Survival score

Example Input:

```

1
68
```

```
24
```

Since, the initial Survival Score was 100. So, with the above first condition: Age is between 66 - 100: 100 - 10 = 90. According to the second condition, the CT score is between 21 - 25: 90 - 10 = 80. As per third condition: Vaccinated Status is 1: 80 + 2 = 82. So final Survival Score =  82

Final Output:

```

Survival Score: 82
```

Question 2:

Generate boarding pass for the passengers of a ship which starts from Chennai  to Andaman. The boarding pass must be generated automatically with a pass number that begins with “CA” and followed by a number that is automatically incremented from value ‘x’, details like passenger name, age, mobile number, address, date of journey and fare. There is a seasonal discount based on the age of the passengers. Write a non member function called discount which calculates the discount in the fare for the passenger with the following discounts. For the age group `between 12 and 58, both inclusive’  there is 30% discount in the fare,  for the age group ‘above 58’, there is 50% discount and for the children (age under 12), 60% discount. Write a C++ program to generate pass for ‘n’ users.

Input Format:

Passenger name

Value of ‘x’

Age
Address
date_of_Journey
mobile number
Original Fare

Output Format:

passenger name

Boarding pass number
age
date_of_Journey
mobile number

Total fare after discount based on age

Example Input:
```
CA
10
39
NAK
10/10/2020
8794000500
5000
```

Final Output:
```
CA
CA10
39
10/10/2020
8794000500
3500
```

Question 3: Given a number ‘n’ and a position ‘p’, write an algorithm and the subsequent ‘C’ program to check if the ‘p-th’ digit from the  leftmost position  of ‘n’ is odd or even. For example, if ‘n’ is 3145782 and p is 4 then you have to check if 5 is odd or even. Since it is odd print ‘Odd’. Make your code accept numbers of larger size.

Input Format:

The first line contains the number, n

The second line contains the position, p

Output Format:

Print either “Odd” or “Even”

Example Input:
```
38769
3
```

Final Output:
```
Odd
```

Question 4:

```
#include <iostream>
using namespace std;
class hiding{
private:
    int num;
    char ch;
public:
    void set(int n, char c) {
        num = n;
        ch = c;
    }
    void get() {
        cout<<"Numbers is: "<<num<< endl;
        cout<<"Char is: "<<ch<<endl;
    }
};
int main(){
    hiding obj;
    obj.set(100, 'X');
    obj.get();
    return 0;
}
```

Practice 5:

```
#include <iostream>
using namespace std;
class hiding{
private:
    int roll;
    char ch;
    int marks;
public:
    void set(int r, char c) {
        roll = r;
        ch = c;
    }
    void get() {
        cout<<"Roll No.: "<<roll<< endl;
        cout<<"Name: "<<ch<<endl;
    }
    void grade(int per){
        marks = per;
        if(per<=100&&per>=80)
            cout<<"Grade: A";
        else if(per<80&&per>=70)
            cout<<"Grade: B";
        else if(per<70&&per>=50)
            cout<<"Grade: C";
        else
            cout<<"Grade: F";
    }
};
int main(){
    hiding obj;
    obj.set(100, 'X');
    obj.get();
    obj.grade(92);
    return 0;
}
```

Question 6:

```
#include <iostream>
using namespace std;
class Sum{
    private:
        int x, y, z;
    public:
    void add(){
        cout<<"Enter two numbers: ";
        cin>>x>>y;
        z= x+y;
        cout<<"Sum of two number is: "<<z<<endl;
    }
};
int main(){
    Sum sm;
    sm.add();
    return 0;
}
```

Practice 7:

```
#include <iostream>
```

```
using namespace std;
class Largest{
    private:
        int x, y, z;
    public:
        int num1, num2, num3;
    void set(){
        cout<<"Enter three numbers: "<<endl;
        cin>>x>>y>>z;
        num1 = x;
        num2 = y;
        num3 = z;
    }
    void get(){
        cout<<"First number: "<<num1<<endl;
        cout<<"Second number: "<<num2<<endl;
        cout<<"Three number: "<<num3<<endl;
    }
    void maxnum(){
        if(num1 > num2 && num1 > num3){
            cout<<"First number is the largest";
        }
        else if(num2 > num1 && num2 > num3){
            cout<<"Second number is the largest";
        }
        else{
            cout<<"Third number is the largest";
        }
    }
};
int main(){
    Largest ls;
    ls.set();
    ls.get();
    ls.maxnum();
    return 0;
}
```

Question 8:

```
#include <iostream>
#include<math.h>
using namespace std;
int main()
```

```
{
  int n = 4;
  int power = 3;
  int result = pow(n,power);
  cout << "Cube of n is : " <<result;
  return 0;
}
```

Practice 9:

```
#include <iostream>
#include <math.h>
using namespace std;

void checkperfectsquare(int n)
{
   if (ceil(sqrt(n)) == floor(sqrt(n))) {
      cout << "Perfect square";
   }
   else {
      cout << "Not a perfect square";
   }
}

int main()
{
   int n = 50;
   checkperfectsquare(n);
   return 0;
}
```

Question 10:

An interface describes the behavior or capabilities of a C++ class without committing to a particular implementation of that class.

The C++ interfaces are implemented using abstract classes and these abstract classes should not be confused with data abstraction which is a concept of keeping implementation details separate from associated data.

```
#include <iostream>
using namespace std;
```

```
class Shape {
  public:
   // pure virtual function providing interface framework.
   //A pure virtual function is a virtual function in C++
   //for which we need not to write any function definition and only we have to declare it.
   //It is declared by assigning 0 in the declaration.
   virtual int getArea() = 0;
   void setWidth(int w) {
      width = w;
   }

   void setHeight(int h) {
      height = h;
   }

   protected:
      int width;
      int height;
};

class Rectangle: public Shape {
  public:
     int getArea() {
        return (width * height);
     }
};

class Triangle: public Shape {
  public:
     int getArea() {
        return (width * height)/2;
     }
};

int main(void) {
  Rectangle Rect;
  Triangle  Tri;

  Rect.setWidth(5);
  Rect.setHeight(7);

  cout << "Total Rectangle area: " << Rect.getArea() << endl;

  Tri.setWidth(5);
  Tri.setHeight(7);

  cout << "Total Triangle area: " << Tri.getArea() << endl;

  return 0;
}
```

Question 11 : The area is the two-dimensional amount of space that an object occupies. Area is measured along the surface of an object and has dimensions of length squared; for example, square feet of material, or centimetres squared.

The area of a rectangle is equal to the height h times the base b; A = h * b

The equation for the area of a trapezoid is one half the sum of the top t and bottom b times the height h; A = h * [ t + b ] / 2

The area of a circle is A = pi * r2, where pi = 3.14 and r = radius.

Develop a program in C++ using function overloading for computing the area of a rectangle, a trapezoid and a circle by a common function name ComputeArea() with different signatures. Assume pi = 3.14. Print only two decimal places for all areas.

Note:

To print only two decimal places of a variable ‘a’, do the following:

```
#include
```

```
cout<<fixed<<setprecision(2)<<a;
```

Input Format:

Read the base and height of a rectangle.

Read the top, bottom and height of a trapezoid.

Read the radius of a circle.

Output Format:

Display the area of a rectangle, trapezoid and circle each in one line

Boundary Conditions:

You can give any valid integer or float values for inputs.

Test Case 1:

```
10 5

10 10 4

5
```

Output:

```
50

70

49.30
```

Test Case 2:

```
12 7

12 12 6

7
```

Output:

```
84
```

108

69.07

Question 12: C++ Program to store GPA of n number of students and display average of it where n is the number of students entered by the user using new and delete Operator for Arrays. The average should be printed till 2 decimal points.

Input:

Read the n value for the number of students.

Read the GPA of n number of students.

Output:

Display the average GPA of the class.

Test Case 1:

5

3 3.4 3.6 3.2 2.3

Output:

3.1

Test Case 1:

4

3 3.9 3.4 3

3.33

Question 12:

```
#include <iostream>
using namespace std;
class hiding{
private:
    int num;
    char ch;
public:
    void set(int n, char c) {
        num = n;
        ch = c;
    }
    void get() {
        cout<<"Numbers is: "<<num<< endl;
        cout<<"Char is: "<<ch<<endl;
    }
};
int main(){
    hiding obj;
    obj.set(100, 'X');
    obj.get();
    return 0;
}
```

Practice 13:

```
#include <iostream>
using namespace std;
class hiding{
private:
    int roll;
    char ch;
    int marks;
public:
    void set(int r, char c) {
        roll = r;
        ch = c;
    }
    void get() {
        cout<<"Roll No.: "<<roll<< endl;
        cout<<"Name: "<<ch<<endl;
    }
    void grade(int per){
        marks = per;
        if(per<=100&&per>=80)
            cout<<"Grade: A";
        else if(per<80&&per>=70)
            cout<<"Grade: B";
        else if(per<70&&per>=50)
            cout<<"Grade: C";
        else
            cout<<"Grade: F";
    }
};
int main(){
    hiding obj;
    obj.set(100, 'X');
    obj.get();
    obj.grade(92);
    return 0;
}
```

Question 14:

```
#include <iostream>
using namespace std;
class Sum{
    private:
        int x, y, z;
    public:
    void add(){
        cout<<"Enter two numbers: ";
        cin>>x>>y;
        z= x+y;
        cout<<"Sum of two number is: "<<z<<endl;
    }
};
int main(){
    Sum sm;
    sm.add();
    return 0;
}
```

Practice 2:

```
#include <iostream>
using namespace std;
class Largest{
    private:
        int x, y, z;
    public:
        int num1, num2, num3;
    void set(){
        cout<<"Enter three numbers: "<<endl;
        cin>>x>>y>>z;
        num1 = x;
        num2 = y;
        num3 = z;
    }
    void get(){
        cout<<"First number: "<<num1<<endl;
        cout<<"Second number: "<<num2<<endl;
        cout<<"Three number: "<<num3<<endl;
    }
    void maxnum(){
        if(num1 > num2 && num1 > num3){
            cout<<"First number is the largest";
        }
        else if(num2 > num1 && num2 > num3){
            cout<<"Second number is the largest";
        }
        else{
            cout<<"Third number is the largest";
        }
    }
};
int main(){
    Largest ls;
    ls.set();
    ls.get();
    ls.maxnum();
    return 0;
}
```

Question 15:

```
#include <iostream>
#include<math.h>
```

```
using namespace std;
int main()
{
  int n = 4;
  int power = 3;
  int result = pow(n,power);
  cout << "Cube of n is : " <<result;
  return 0;
}
```

Practice 16:

```
#include <iostream>
#include <math.h>
using namespace std;

void checkperfectsquare(int n)
{
   if (ceil(sqrt(n)) == floor(sqrt(n))) {
      cout << "Perfect square";
   }
   else {
      cout << "Not a perfect square";
   }
}

int main()
{
   int n = 50;
   checkperfectsquare(n);
   return 0;
}
```

Question 17:

An interface describes the behavior or capabilities of a C++ class without committing to a particular implementation of that class.

The C++ interfaces are implemented using abstract classes and these abstract classes should not be confused with data abstraction which is a concept of keeping implementation details separate from associated data.

```
#include <iostream>
```

```
using namespace std;

class Shape {
   public:
    // pure virtual function providing interface framework.
    //A pure virtual function is a virtual function in C++
    //for which we need not to write any function definition and only we have to declare it.
    //It is declared by assigning 0 in the declaration.
    virtual int getArea() = 0;
    void setWidth(int w) {
        width = w;
    }

    void setHeight(int h) {
        height = h;
    }

    protected:
        int width;
        int height;
};

class Rectangle: public Shape {
   public:
      int getArea() {
         return (width * height);
      }
};

class Triangle: public Shape {
   public:
      int getArea() {
         return (width * height)/2;
      }
};

int main(void) {
   Rectangle Rect;
   Triangle  Tri;

   Rect.setWidth(5);
   Rect.setHeight(7);

   cout << "Total Rectangle area: " << Rect.getArea() << endl;
```

```
    Tri.setWidth(5);
    Tri.setHeight(7);

    cout << "Total Triangle area: " << Tri.getArea() << endl;

    return 0;
}
```

Question 18:

```
#include <iostream>
using namespace std;

class Shape
{
public:
    Shape(int l, int w)
    {
        length = l;
        width = w;
    }    void get_Area()
    {
        cout << "This is call to parent class area" << endl;
    }

protected:
    int length, width;
};

class Square : public Shape
{
public:
    Square(int l = 0, int w = 0)
        : Shape(l, w)
    {
    }    int get_Area()
    {
        cout << "Square area: " << length * width << endl;
        return (length * width);
    }
};
class Rectangle : public Shape
```

```
{
public:
    Rectangle(int l = 0, int w = 0)
        : Shape(l, w)
    {
    }    int get_Area()
    {
        cout << "Rectangle area: " << length * width
             << endl;
        return (length * width);
    }
};

int main(void)
{
    Shape* s;
    Square sq(5, 5);
    Rectangle rec(4, 5);
    s = &sq;
    s->get_Area();
    s = &rec;
    s->get_Area();

    return 0;
}
```

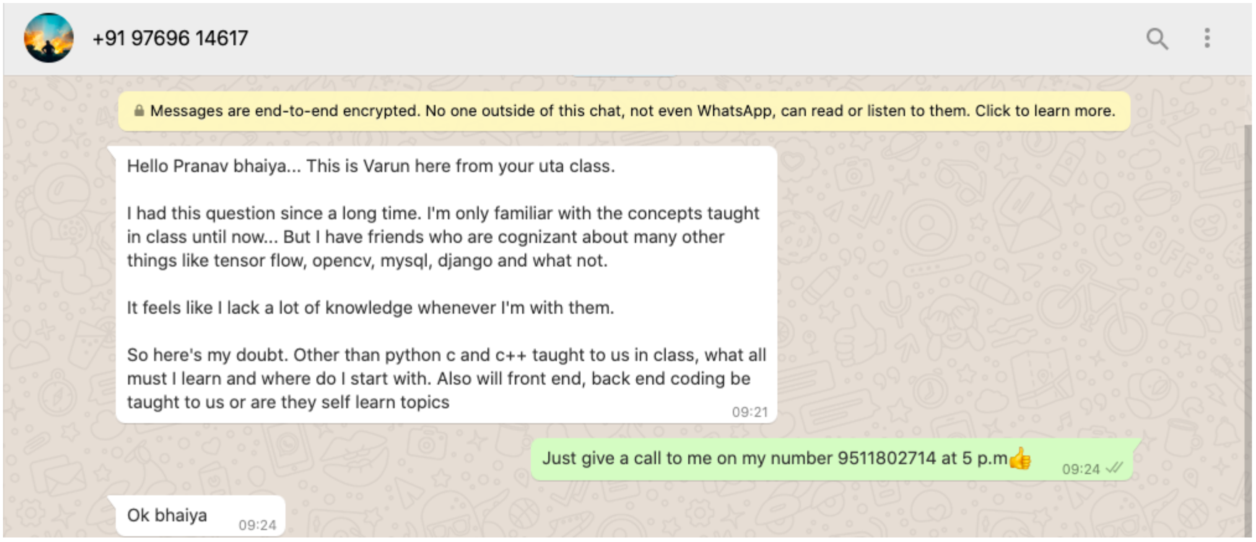
## Question 19: Placement Top 50 Problems

Question 1. What Is Oops?
Question 2. Write Basic Concepts Of Oops?
Question 3. What Is A Class?
Question 4. What Is An Object?
Question 5. What Is Encapsulation?
Question 6. What Is Polymorphism?
Question 7. What Is Inheritance?
Question 8. What Are Manipulators?
Question 9. Define A Constructor?
Question 10. Define Destructor?
Question 11. What Is Inline Function?
Question 12. What Is A Virtual Function?
Question 13. What Is Friend Function?
Question 14. What Is Function Overloading?

Question 15. What Is Operator Overloading?
Question 16. What Is An Abstract Class?
Question 17. What Is A Ternary Operator?
Question 18. What Is The Use Of Finalize Method?
Question 19. What Are Different Types Of Arguments?
Question 20. What Is Super Keyword?
Question 21. What Is Method Overriding?
Question 22. What Is An Interface?
Question 23. What Is Exception Handling?
Question 24. What Are Tokens?
Question 25. Difference Between Overloading And Overriding?
Question 26. Difference Between Class And An Object?
Question 27. What Is An Abstraction?
Question 28. What Are Access Modifiers?
Question 29. What Is Sealed Modifiers?
Question 30. How Can We Call The Base Method Without Creating An Instance?
Question 31. What Is The Difference Between New And Override?
Question 32. What Are The Various Types Of Constructors?
Question 33. What Is Early And Late Binding?
Question 34. What Is ‘this’ Pointer?
Question 35. What Is The Difference Between Structure And A Class?
Question 36. What Is The Default Access Modifier In A Class?
Question 37. What Is Pure Virtual Function?
Question 38. What Are All The Operators That Cannot Be Overloaded?
Question 39. What Is Dynamic Or Run Time Polymorphism?
Question 40. Do We Require Parameter For Constructors?
Question 41. What Is A Copy Constructor?
Question 42. What Does The Keyword Virtual Represented In The Method Definition?
Question 43. What Are Base Class, Sub Class And Super Class?
Question 44. What Is Static And Dynamic Binding?
Question 45. How Many Instances Can Be Created For An Abstract Class?
Question 46. Which Keyword Can Be Used For Overloading?
Question 47. What Is The Default Access Specifier In A Class Definition?
Question 48. Which Oops Concept Is Used As Reuse Mechanism?
Question 49. Which Oops Concept Exposes Only Necessary Information To The Calling Functions?
Question 50. What Are The Types Of Constructors?

**FEEDBACK FROM STUDENTS**

Sample Feedback collected from the students in the form of email / snapshot of the handwritten feedback (with their sign)



+91 97696 14617

Messages are end-to-end encrypted. No one outside of this chat, not even WhatsApp, can read or listen to them. Click to learn more.

Hello Pranav bhaiya... This is Varun here from your uta class.

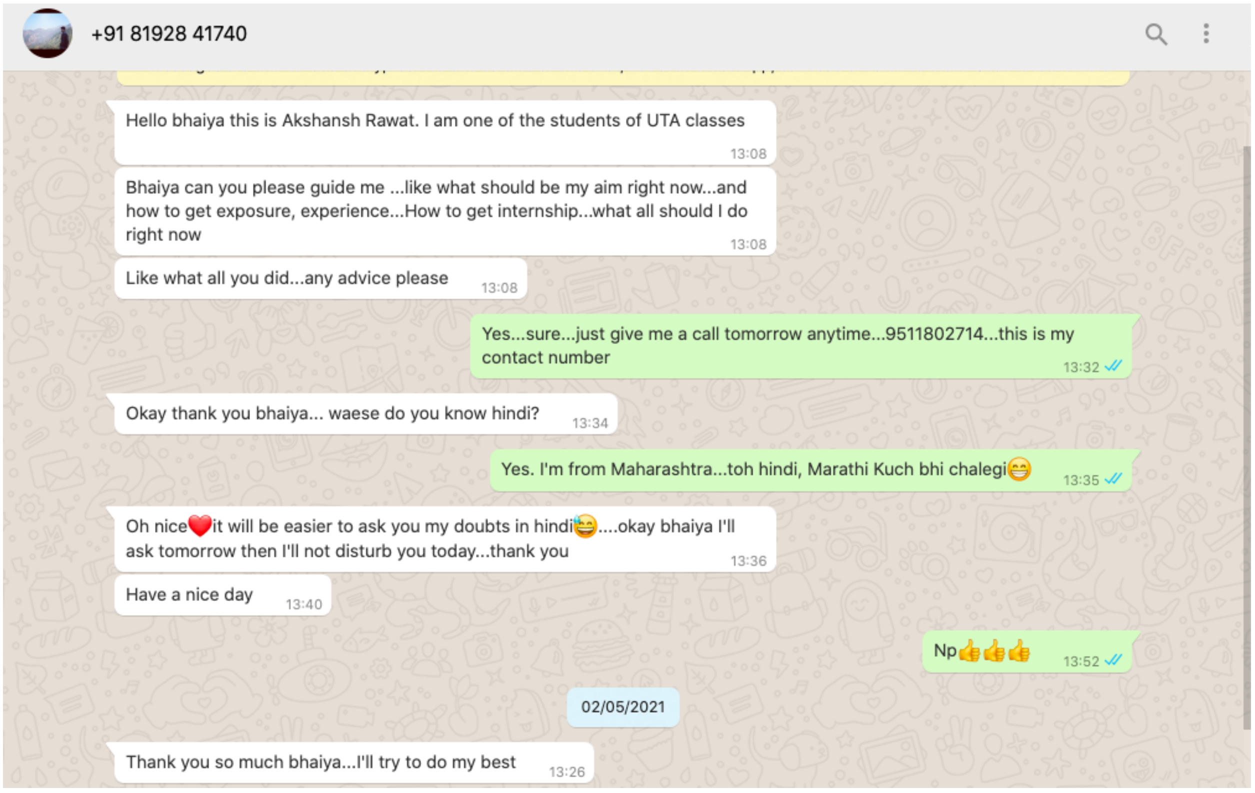
I had this question since a long time. I'm only familiar with the concepts taught in class until now... But I have friends who are cognizant about many other things like tensor flow, opencv, mysql, django and what not.

It feels like I lack a lot of knowledge whenever I'm with them.

So here's my doubt. Other than python c and c++ taught to us in class, what all must I learn and where do I start with. Also will front end, back end coding be taught to us or are they self learn topics 09:21

Just give a call to me on my number 9511802714 at 5 p.m👍 09:24

Ok bhaiya 09:24



+91 81928 41740

Hello bhaiya this is Akshansh Rawat. I am one of the students of UTA classes 13:08

Bhaiya can you please guide me ...like what should be my aim right now...and how to get exposure, experience...How to get internship...what all should I do right now 13:08

Like what all you did...any advice please 13:08

Yes...sure...just give me a call tomorrow anytime...9511802714...this is my contact number 13:32

Okay thank you bhaiya... waese do you know hindi? 13:34

Yes. I'm from Maharashtra...toh hindi, Marathi Kuch bhi chalegi😁 13:35

Oh nice❤️it will be easier to ask you my doubts in hindi😅....okay bhaiya I'll ask tomorrow then I'll not disturb you today...thank you 13:36

Have a nice day 13:40

Np👍👍👍 13:52

02/05/2021

Thank you so much bhaiya...I'll try to do my best 13:26