LOVELY PROFESSIONAL UNIVERSITY

ACADEMIC TASK 1

CSE101 : COMPUTER PROGRAMMING

INCOME TAX CALCULATOR(MINI PROJECT)

TEAM MEMBERS -

NAME	ROLL NUMBER	REGISTRATION
ABHINAV GANGWAR	RKOCCKB37	12216905
SHUBHAM K. RAI	RKOCCKB38	12217042
MADHAV ARORA	RKOCCKB39	12217099
PRIYANSH KHARE	RKOCCKB40	12217140

SUBMITTED TO – AMANDEEP KAUR ma'am

ACKNOWLEDEMENT

I would like to take this opportunity to express my sincere gratitude and appreciation to all those who have contributed to the successful completion of this project.Firstly, I would like to thank my supervisor, Amandeep Kaur , for their invaluable guidance, support, and encouragement throughout the project. Their expertise and insights have been instrumental in shaping the direction and scope of this work.I would also like to extend my thanks to Shubham Kumar Rai, Madhav Arora, and Priyansh Khare for their assistance and support in various aspects of the project. Their contributions have been invaluable in helping me to overcome challenges and achieve the project's objectives.Finally, I would like to express my gratitude to the institution and department for providing me with the resources and facilities necessary to complete this project.Once again, thank you to everyone who has contributed to this project. Your support and encouragement have been greatly appreciated.

INCOME TAX CALCULATOR

An income tax calculator is a tool that helps individuals estimate the amount of federal, state, and local taxes they owe based on their income. The calculator takes into account various factors such as filing status, deductions, and credits to provide an accurate estimate of the taxes owed.

The federal income tax calculator is a popular tool that helps individuals calculate their federal income tax liability. It takes into account the latest tax laws and regulations to provide an accurate estimate of the taxes owed. The calculator requires individuals to input their income, filing status, and deductions to calculate the taxes owed.

The IRS also provides a tax withholding estimator that helps individuals estimate the correct amount of tax to withhold from each paycheck. This tool helps ensure that individuals do not owe more in taxes than they can afford to pay. The estimator requires individuals to input their most recent pay stubs, their most recent income tax return, and other relevant information to calculate the correct withholding amount.

Overall, income tax calculators are useful tools that help individuals estimate their tax liability and ensure that they are paying the correct amount of taxes. They are easy to use and can provide valuable insights into an individual's tax situation.

MODULES USED

1. ADD NEW RECORD MODULE

In this module the program will ask user to enter their details of income and deductions. It will create a record of the new user.

```
void getPersonDetails(){
    printf("Enter details\n\n");
    struct Taxpayor person;
    printf("Enter your name : ");
    getchar();
    scanf("%[^\n]%*c",person.name);
    printf("Income from salary : ");
    scanf("%f",&person.income[0]);
    printf("Exempt Allowance : ");
    scanf("%f",&person.income[1]);
    printf("Income from interests : ");
    scanf("%f",&person.income[2]);
    printf("Interest paid on house loan : ");
    scanf("%f",&person.income[3]);
    printf("Rental Income received : ");
    scanf("%f",&person.income[4]);
    printf("Interest paid on loan : ");
    scanf("%f",&person.income[5]);
    printf("Income from digital assets : ");
    scanf("%f",&person.income[6]);
    printf("Income from other sources : ");
    scanf("%f",&person.income[7]);
    printf("-------------------------------------------------\n");
    printf("Basic deduction (80C) : ");
    scanf("%f",&person.deduction[0]);
    printf("Interests from deposits (80TTA) : ");
    scanf("%f",&person.deduction[1]);
    printf("Medical insurance (80D) : ");
    scanf("%f",&person.deduction[2]);
    printf("Donation to charity (80G) : ");
    scanf("%f",&person.deduction[3]);
    printf("Interest on Educational loan (80E) : ");
    scanf("%f",&person.deduction[4]);
    printf("Interest on housing loan (80EEA) : ");
    scanf("%f",&person.deduction[5]);
    printf("Contribution to NPS (80CCD) : ");
    scanf("%f",&person.deduction[6]);
    if(person.deduction[0]>150000)person.deduction[0] = 150000;
    if(person.deduction[1]>10000)person.deduction[1] = 10000;
```

Here you can see it asks for every detail form the user. After taking the input from the user this module will rectify the given data according to the limits provided by the Indian Government.

```
if(person.deduction[1]>10000)person.deduction[1] = 10000;
if(person.deduction[2]>25000)person.deduction[2] = 25000;
if(person.deduction[5]>150000)person.deduction[5] = 150000;
if(person.deduction[6]>50000)person.deduction[6] = 50000;
for (int i = 0; i <= 7; i++)person.totalIncome += person.income[i];
for (int i = 0; i < 7; i++)person.totalDeduction += person.deduction[i];
person.taxableIncome = person.totalIncome - person.totalDeduction;
float tax_=0;
if(person.taxableIncome <= 5000000){
    if(person.taxableIncome<=300000){
        tax_ += 0;
        person.taxableIncome=0;
    }
    else{
        if(person.taxableIncome>300000){
            if((person.taxableIncome-300000)>=300000)tax_ += 15000;
            else tax_ += (person.taxableIncome-300000)*0.05;
        }
        if(person.taxableIncome>600000){
            if((person.taxableIncome-600000)>=300000)tax_ += 30000;
            else tax_ += (person.taxableIncome-600000)*0.1;
        }
        if(person.taxableIncome>900000){
            if((person.taxableIncome-900000)>=300000)tax_ += 45000;
            else tax_ += (person.taxableIncome-900000)*0.15;
        }
        if(person.taxableIncome>1200000){
            if((person.taxableIncome-1200000)>=300000)tax_ += 60000;
            else tax_ += (person.taxableIncome-1200000)*0.2;
        }
        if(person.taxableIncome>1500000)tax_ += (person.taxableIncome-1500000)*0.3;
        if(person.taxableIncome==1500000)tax_ = 150000;
    }
}
else {
    if(person.taxableIncome>5000000){
        if((person.taxableIncome-5000000)>5000000)tax_ += 500000;
        else tax_ += (person.taxableIncome-5000000)*0.1;
    }
```

```
else {
    if(person.taxableIncome>5000000){
        if((person.taxableIncome-5000000)>5000000)tax_ += 500000;
        else tax_ += (person.taxableIncome-5000000)*0.1;
    }
    if(person.taxableIncome>10000000){
        if((person.taxableIncome-10000000)>=10000000)tax_ += 1500000;
        else tax_ += (person.taxableIncome-10000000)*0.15;
    }
    if(person.taxableIncome>20000000){
        if((person.taxableIncome-30000000)>=30000000)tax_ += 7500000;
        else tax_ += (person.taxableIncome-5000000)*0.25;
    }
    if(person.taxableIncome>50000000){
        tax_ += (person.taxableIncome-50000000)*0.37;
    }
}
tax_ += tax_*0.4;//APPLYING CESS
person.tax = tax_;
taxpayors[newEntry] = person;
newEntry++;
printf("----------------------------------------------------------------\n");
printf("Taxable Income : %f",person.taxableIncome);
printf("\nPayable Tax : %f",person.tax);
printf("\n----------------------------------------------------------------\n");
```

As shown in the snapshot the module is calculating the total taxable income. Now it will calculate the tax according to income slabs provided by the Indian Gov. and will display the TAXABLE INCOME and PAYABLE TAX to the user after storing their record.

2. LIST ALL THE TAXPAYERS WITH INCOME TAX TO BE PAYED MODULE

In this module, all the information about the taxpayers will be displayed.

```
void list_taxpayor(){
    if(newEntry==0)printf("No record found.\n");
    else for(int i=0;i<newEntry;i++)printf("Name : %s\nTaxable Income : %.2f\nPayable Tax : %.2f",taxpayors[i].name,taxpayors[i].
    taxableIncome,taxpayors[i].tax);
}
```

Here if there is no record then a message will be displayed “No record found.” else list of all taxpayers with their name, taxable income and payable tax will be displayed.

3. SEARCH MODULE

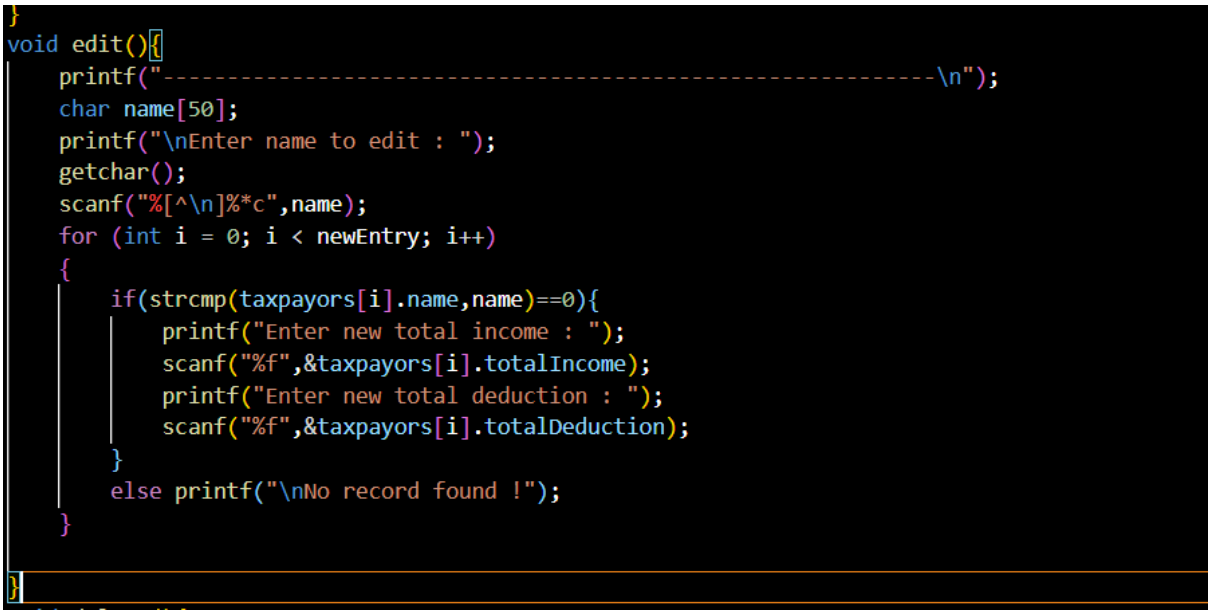
In this module search operation will be performed.

```
void search(){
    printf("------------------------------------------------------------\n");
    char name[20];
    printf("\nEnter name : ");
    getchar();
    scanf("%[^\n]%*c",name);
    for(int i=0;i<newEntry;i++){
        if(strcmp(taxpayors[i].name,name)==0){
            printf("Name : %s\nTaxable Income : %.2f\nPayable Tax : %.2f",taxpayors[i].name,taxpayors[i].taxableIncome,taxpayors[i].tax);
            return;
        }
    }
    printf("Not found!\n");
}
```

Here, name of the user is asked and if it is present in the record, then their name with taxable income and payable tax is displayed on the terminal. Else a message “Not found!” is shown on the terminal.

4. EDIT MODULE

This module gives the user to edit their data which was given.



```
}
void edit(){
    printf("------------------------------------------------------------\n");
    char name[50];
    printf("\nEnter name to edit : ");
    getchar();
    scanf("%[^\n]%*c",name);
    for (int i = 0; i < newEntry; i++)
    {
        if(strcmp(taxpayors[i].name,name)==0){
            printf("Enter new total income : ");
            scanf("%f",&taxpayors[i].totalIncome);
            printf("Enter new total deduction : ");
            scanf("%f",&taxpayors[i].totalDeduction);
        }
        else printf("\nNo record found !");
    }

}
```

Here user's name is asked to perform the search operation. Then if it present in the record then their new income and deductions will be asked and is stored in the record with their tax which is calculated by the program as per the new data given by the user. In case if the name of the user is not present in the record then a message is shown "No record found !" on the terminal.

5. DELETE MODULE

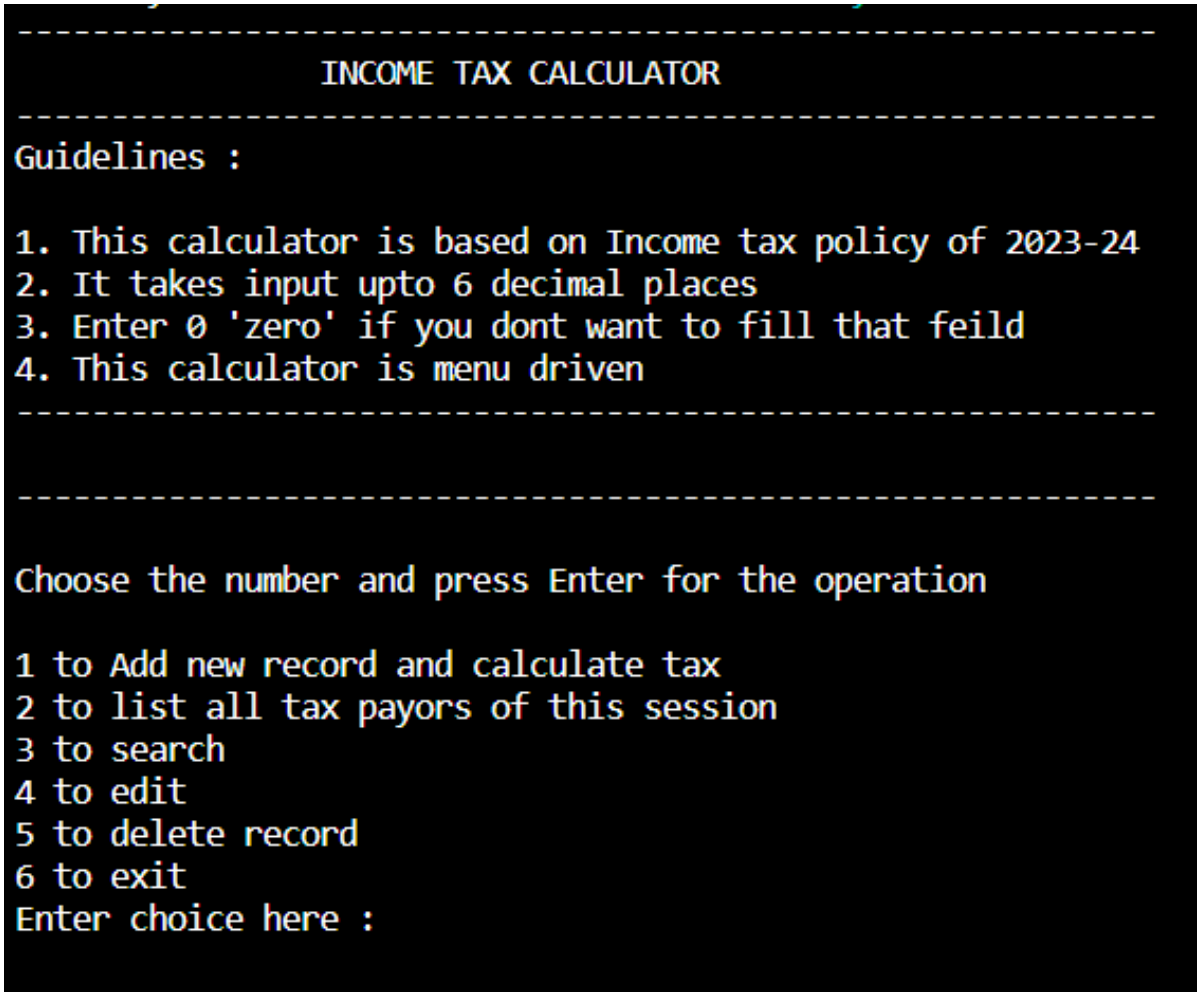
This module allows the user to remove their data from the record.

```
void delete(){
    char name[50];
    printf("------------------------------------------------------------\n");
    printf("\nEnter the name to be deleted : ");
    getchar();
    scanf("%[^\n]%*c",name);
    for(int i=0;i<newEntry;i++){
        if(strcmp(taxpayors[i].name,name)==0){
            for(int j =i;j<newEntry-1;j++){
                taxpayors[j] = taxpayors[j+1];
                newEntry--;
            }
            printf("Successfully Deleted !");
            return;
        }
        else printf("No data found!");
    }
}
```

In this module, name of the user is asked and if this name is present in the records then it will remove it from there and a message is displayed “Successfully Deleted !”. In case user entered a name which is not there in the records then a message is shown “No data found!” on the terminal.

WORKING OF PROGRAM

The program starts with Income Tax display with guidelines of the program of how it will work.



This is a menu driven program and user has to choose the operation.

CASE I - WHEN USER CHOSE 1

Then the program will ask you about personal details your name, incomes and deductions.

```
Enter choice here : 1
Enter details

Enter your name : Abhinav
Income from salary : 2022000
Exempt Allowance : 0
Income from interests : 0
Interest paid on house loan : 0
Rental Income received : 0
Interest paid on loan : 0
Income from digital assets : 0
Income from other sources : 20000
-------------------------------------------------
Basic deduction (80C) : 0
Interests from deposits (80TTA) : 12000
Medical insurance (80D) : 9000
Donation to charity (80G) : 0
Interest on Educational loan (80E) : 0
Interest on housing loan (80EEA) : 0
Contribution to NPS (80CCD) : 0
-----------------------------------------------------------
Taxable Income : 2023000.000000
Payable Tax : 429660.000000
-----------------------------------------------------------
```

When you enter all the asked details correctly then it will generate the taxable income with payable tax and display it on the terminal. User's record is added automatically when this operation is done.

CASE II - WHEN USER COOSE 2

List of all taxpayers is displayed to the terminal.

```
Enter choice here : 2
Name : Timmy
Taxable Income : 1005000.00
Payable Tax : 85050.00

Name : Johnny
Taxable Income : 0.00
Payable Tax : 0.00
```

here details of two taxpayers is shown.

CASE III - WHEN USER CHOSE 3

User is allowed to search the details by name.

```
Enter choice here : 3
------------------------------------------------------------

Enter name : Johnny
Name : Johnny
Taxable Income : 0.00
Payable Tax : 0.00
```

Details of the searched user is shown

If user entered a name which is not in the record.

```
Enter choice here : 3
------------------------------------------------------------

Enter name : Isah
Not found!
```

Then “Not found!” is displayed.

CASE IV - WHEN USER CHOSE 4

Users are allowed to edit the previously entered details. They are asked to enter the name to search their details in record and then option to edit will be shown.

```
Enter choice here : 4
------------------------------------------------------------

Enter name to edit : Johnny
Enter new total income : 2000000
Enter new total deduction : 20000
Updated Successfully!
```

In case users enter an invalid name which is not there in the records then a message “Entered name is invalid!” is displayed on the terminal.

```
Enter choice here : 4
------------------------------------------------------------

Enter name to edit : Johhny
Entered name is invalid!
```

CASE V - WHEN USER CHOSE 5

Users are allowed to delete their entry from the records. They are asked their name and the data is removed from the records.

```
Enter choice here : 5
------------------------------------------------------------

Enter the name to be deleted : Johnny
Successfully Deleted !
```

A message is displayed after deleting the record.

CASE VI - WHEN USER CHOSE 6

When user chose this option then they will be taken out from the calculator program by showing a message “Successfully exited !” .

```
Enter choice here : 6
Successfully Exited !
```

CONCLUSION

In conclusion, the project of creating an income tax calculator using the C programming language has been successfully completed. The program is designed to calculate the income tax owed based on the user's income and the tax brackets for the current year. The program takes into account the progressive tax system in India, which means that higher income levels are taxed at higher rates.

The program uses a set of step-by-step instructions to direct the computer to perform the necessary calculations and produce the desired results. The program is user-friendly and easy to use, requiring the user to input their income and filing status to calculate the taxes owed.

The project has provided valuable insights into the process of creating a program that performs complex calculations and produces accurate results. It has also demonstrated the importance of understanding the tax system and the various tax brackets and rates that apply to different income levels.

Overall, the project has been a valuable learning experience, providing an opportunity to apply the principles of computer programming to a real-world problem. The income tax calculator program is a useful tool that can help individuals estimate their tax liability and ensure that they are paying the correct amount of taxes.