**ACKNOWLEDGMENT**

I would like to acknowledge the guidance and constructive suggestions provided by the teachers and individuals in course of the preparation of this lab project. I would like to express our sincere thanks to **Mr.RajaramThapa**, computer instructor for his assistance and help during the process.

I would like to convey my thanks to all teaching and non-teaching staff members of St.Lawrence college for their invaluable support and help throughout the project period.

Also, I would like to express my deep sense of gratitude to my friend **Mr.Devraj silwal, Mr.Abishek khadka, Mr.Ashim lamsal** all my beloved friends and family members who tried to encourage me while doing this project with their valuable suggestions to approach our goals. I am thankful to all the peoples who have either directly ot indirectly helped for the successful completion of the project.

Lastly, I would humbly welcome any suggestions and guidance For the improvement of this lab work.

Thank You.

**OBJECTIVE**

This report is for the partial fulfillment of the requirement for grade 12 computer science course designed by NEB. The main objective of preparing this report is to gain the knowledge of c programming and analyzing the output of respective program. The report is submitted to the head of department of computer science for the practical examination purpose. This lab report is for analyzing and solving skill related to c programming.

**INRODUCTION**

C PROGRAMMING is considered as a system programming language. It is also a structured programming language which is used to develop operating system, business system, word processing , database management system etc. C was an offspring of the Basics Combined Programming Language developed in 1960’s at Cambridge University. It is standardized programming language for ANSIC is standard programming language. It is system independent programming language.

**Advantages of C-programming language:**

* C language is a structured programming language which allows user to think of a problem in terms of function modules or blocks.
* Due to the availability of large number of functions, the programming task becomes simple.
* C language is a building block for many other currently known languages.

**Disadvantages of C-programming language:**

* C doesn’t have the concept of constructor or destructor.
* There is no runtime checking in C language.
* C does not have concept of OOPs, that’s why C++ is developed.
* C doesn’t have the concept of namespace.
* There is no strict type checking. For example, we can pass an integer value.

Title:

# WAP to calculate simple interest.

Source code:

#include<stdio.h>

#include<conio.h>

int main()

{

float p,r,t,i;

printf("Enter the principle:\t");

scanf("%f",&p);

printf("Enter the rate of interest:\t");

scanf("%f",&r);

printf("Enter the time:\t");

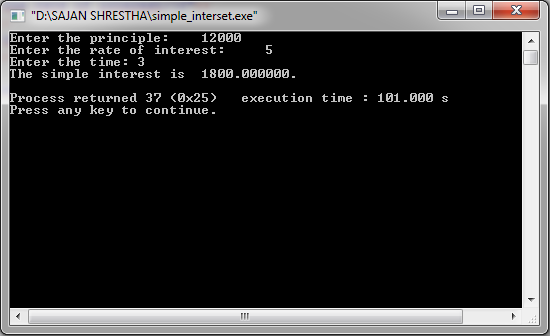
scanf("%f",&t);

i=p\*t\*r/100;

printf("The simple interest is \t%f.\n",i);

}

Output:



Title:

# WAP to convert the given days into years, months and remaining days.

Source code:

#include <stdio.h>

#include <conio.h>

void main()

{

intd,m,y,rem,r;

printf("Enter number of days:\t");

scanf("%d",&d);

y=d/365;

rem=d%365;

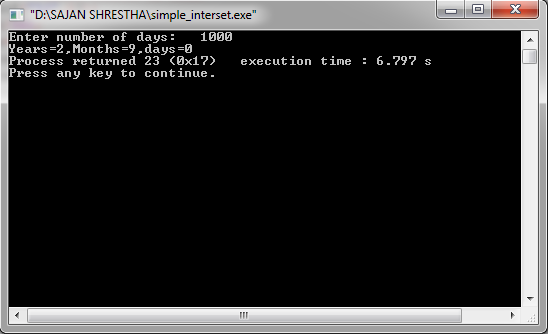
m=rem/30;

r=rem%30;

printf("Years=%d,Months=%d,days=%d",y,m,r);

}

Output:



**CONCLUSION**

From the above assignment, I got to know about the basics of C-programming concept.I learnt to use the basic input output functions like scanf() and printf(). Likewise, I was able to perform the programs like calculating simple interest and converting given daysinto years, months and remaining days.

Title:

# WAP to find the commission amount on the basis of sales amount.

Sales amount (rs) Comission

0-1000 5%

1001-2000 10%

>2000 12%

Source code:

#include<stdio.h>

#include<conio.h>

void main()

{

floatn,c;

printf("Enter the sales amount:\t");

scanf("%f",&n);

if(n<=1000)

{

c=0.05\*n;

printf("the total commission is Rs%f.",c);

}

else if(n>1000&&n<=2000)

{

c=50+0.1\*(n-1000);

printf("the total commission is Rs%f.",c);

}

else if(n>2000)

{

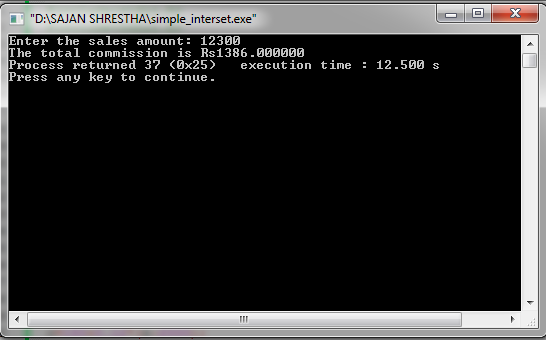
c=150+0.12\*(n-2000);

printf("The total commission is Rs%f",c);

}

}

Output:



Title:

# WAP to perform arithmetic operation on integer’s data using switch statement.

Source code:

#include<stdio.h>

#include<conio.h>

void main()

{

intn,m,t;

char c;

printf("Enter two numbers and operator :\n");

scanf("%d %d %c", &n, &m, &c);

switch(c)

{

case '+':printf("Addition is : %d", n+m);

break;

case '-':printf("Substraction is %d", n-m);

break;

case '\*':printf("Multiplication is %d", n\*m);

break;

case '/':printf("Division is %f", (float)n/m);

break;

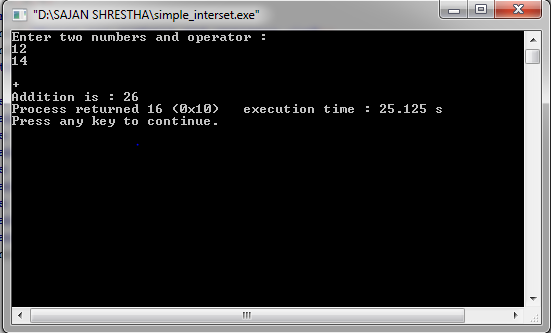
default:

printf("Not valid");

}

}

Output:



**CONCLUSION**

After completing the above assignment, I learnt about the uses of if/else statement while performing the programs including the conditional sentences. Along with this, I also got to know about the functions of ‘switch’ while asking the users about the choice questions. Also, I became able to find the commission on the basis of given information and perform arithmetic operation according to the user’s choice using a C-program.

Title:

# WAP to find sum of first n natural numbers.

Source code:

#include<stdio.h>

#include<conio.h>

void main()

{

inti,n,sum=0;

printf("Enter the number upto which the sum of natural number is needed:");

scanf("%d",&n);

for(i=1;i<=n;i++)

{

sum=sum+i;

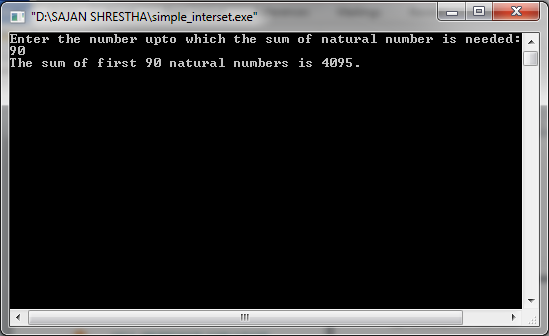
}

printf("The sum of first %d natural numbers is %d.",n,sum);

getch();

}

Output:



Title:

# WAP to find if the entered number is prime or composite.

Source code:

#include<stdio.h>

#include <conio.h>

void main()

{

inti,j,n,c=0;

printf("Enter a number\n");

scanf("%d",&n);

for(i=1;i<=n;i++)

{

j=n%i;

if(j= =0)

{

c++;

}

}

if (c==2)

printf("%d is a prime number.",n);

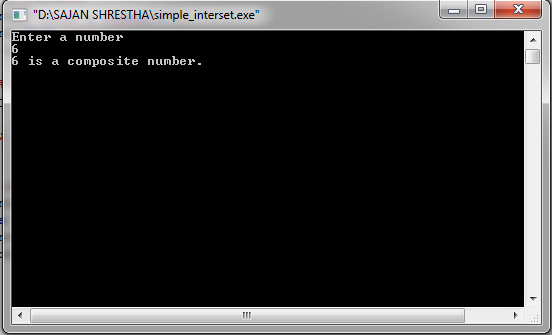
else

printf("%d is a composite number.",n);

getch();

}

Output:



**CONCLUSION**

While performing the above assignment, I got the information about different types of loops like for loop, while loop and do-while loop. Also, I learnt to find the sum of n natural numbers using loop and determine whether the entered number is prime or composite using a C-program.

Title:

# WAP to read the salary of n number of employees and count the number of employee getting salary between 10000 and 40000.

Source code:

#include<stdio.h>

#include<conio.h>

void main()

{

inti,c=0,n,a[100];

printf("Enter the number of employees:");

scanf("%d",&n);

for(i=0;i<n;i++)

{

printf("Enter salary of employee %d:",i+1);

scanf("%d",&a[i]);

}

for(i=0;i<n;i++)

{

if(a[i]>=10000&&a[i]<=40000)

{

c++;

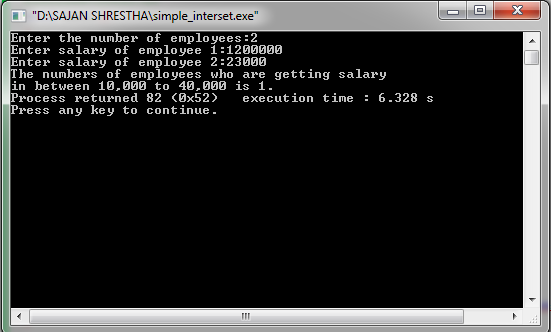
}

}

printf("The numbers of employees who are getting salary in between 10,000 to 40,000 is %d.",c );

}

Output:



Title:

# WAP to input n numbers and search a given number by the user.

Source code:

#include <stdio.h>

#include <conio.h>

void main()

{

int a[100],search,n,i;

printf("Enter how many numbers should be there:\n");

scanf("%d",&n);

for (i = 0; i < n; i++)

{

printf("Enter %d number=",i+1);

scanf("%d", &a[i]);

}

printf("\nEnter the number you want to search:");

scanf("%d", &search);

for (i = 0; i < n; i++)

{

if (a[i] == search)

{

printf("\n%d is present at location %d.\n", search, i+1);

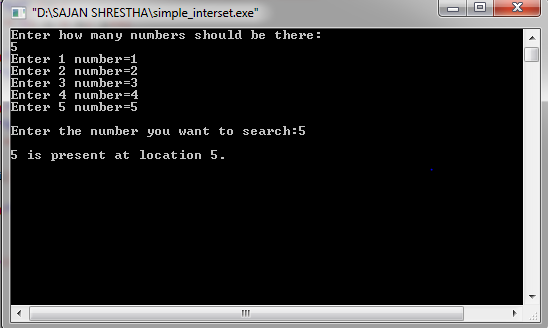
}

}

getch();

}

Output:



Title:

# WAP to find the sum of two matrices.

Source code:

#include<stdio.h>

#include <conio.h>

void main()

{

inti,j,n,a[2][2],b[2][2],c[2][2];

printf("Enter the numbers in matrix A:\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{printf("a%d%d=",i+1,j+1);

scanf("%d",&a[i][j]);

}

printf("\n");}

printf("Enter the numbers in matrix B:\n");

for(i=0;i<2;i++)

{for(j=0;j<2;j++)

{printf("b%d%d=",i+1,j+1);

scanf("%d",&b[i][j]);

}printf("\n");

}

printf("Addition of matrix A and B is:\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{c[i][j]=a[i][j]+b[i][j];

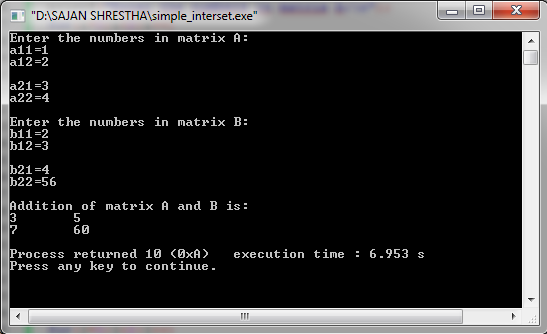
printf("%d",c[i][j]);

printf("\t");}

printf("\n");

}}

Output:



Title:

# WAP to find the transpose of a matrix.

Source code:

#include <stdio.h>

#include <conio.h>

void main()

{

inti,j,n,a[3][3],b[3][3];

printf("Enter the elements of matrix:\n");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

printf("a%d%d=",i+1,j+1);

scanf("%d",&a[i][j]);

}

printf("\n");

}

printf("The Transpose of given matrix is: \n");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

b[i][j]=a[j][i];

printf("%d",b[i][j]);

printf("\t");

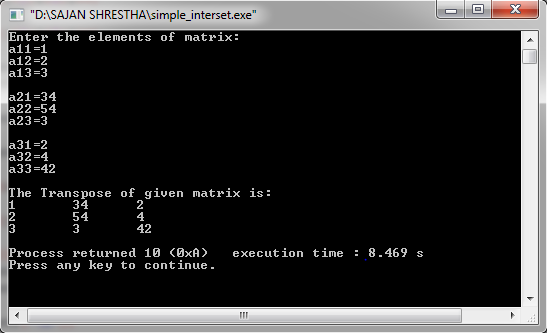
}

printf("\n");

}

}

Output:



**CONCLUSION**

After completing the above assignment, I came to know about what an array is. Along with this I became able to calculate the no. of employees getting salary between a fixed range, locating a number given by the user, performing arithmetic operation like addition, subtraction and finding the transpose of given matrix using array in a C-program.

Title:

# WAP to sort the name of 5 students in Alphabetical order.

Source code:

#include<stdio.h>

#include<string.h>

void main()

{

inti,j;

charstr[100][100],s[100];

printf("Enter names of 5 students:\n");

for(i=0;i<5;i++)

{

scanf("%s",str[i]);

}

for(i=0;i<5;i++)

{

for(j=i+1;j<5;j++)

{

if(strcmp(str[i],str[j])>0)

{

strcpy(s,str[i]);

strcpy(str[i],str[j]);

strcpy(str[j],s);

}}}

printf("The sorted order of names are:\n");

for(i=0;i<5;i++)

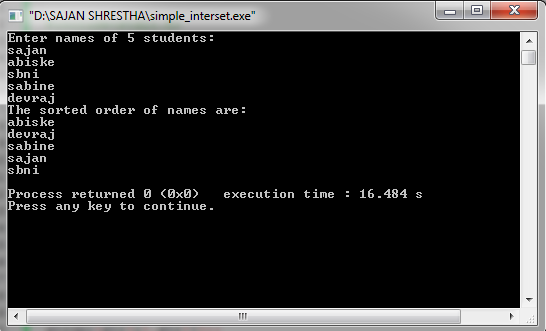
{

printf("%s\n",str[i]);

}

}

Output:



Title:

# WAP to count numbers of vowels and consonants in a line of text.

Source code:

# include <stdio.h>

# include <conio.h>

#include <string.h>

void main()

{

char s[100];

inti,vowel=0,consonant=0;

printf("Enter the line of text:");

gets(s);

for (i=0;s[i];i++)

{

if(s[i]=='a'||s[i]=='e'||s[i]=='i'||s[i]=='o'||s[i]=='u'||s[i]=='A'||s[i]=='E'||s[i]=='I'||s[i]=='O'||s[i]=='U')

vowel++;

else

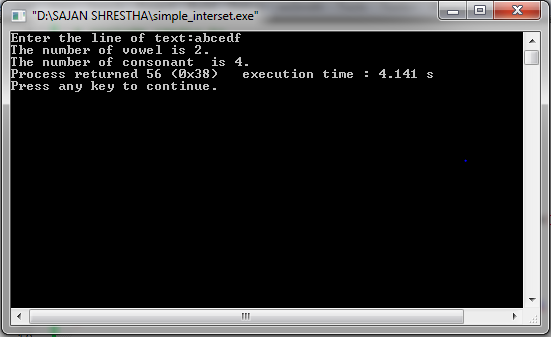
consonant++;

}

printf("The number of vowel is %d.\nThe number of consonant is %d.",vowel,consonant);

}

Output:



**CONCLUSION**

After accomplishing the given assignment on string , I knew about different string handling functions like strlen(), strcat(), strrev(),etc. Likewise, I became able to perform the C-programs like sorting names in alphabetical order and counting the number of vowels and consonants in the given line of text.

Title:

# WAP to find the area and circumference of a circle.

Source code:

#include <stdio.h>

#include <conio.h>

int area(int);

int circumference (int);

void main()

{

intr,a,c;

printf("Enter radius of the circle:");

scanf("%d",&r);

a=area(r);

c=circumference(r);

printf("The area of circle is %d\n The circumference of circle is %d.",a,c);

}

int area(r)

{

int a;

a=3.1415\*r\*r;

return a;

}

int circumference(r)

{

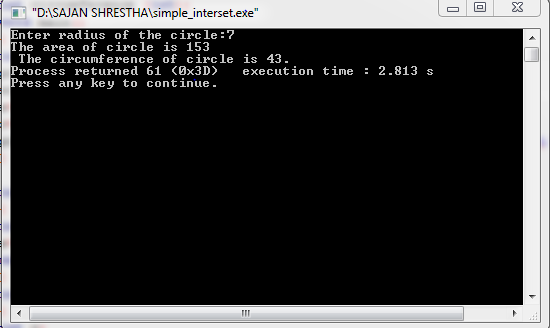
int c;

c=2\*3.1415\*r;

return c;

}

Output:



Title:

# WAP to find the largest and smallest among 10 number.

Source code:

#include<stdio.h>

#include<conio.h>

voidlargest\_and\_smallest();

int main()

{

int a;

largest\_and\_smallest();

}

voidlargest\_and\_smallest()

{int a[10],i,j,k;

printf("Enter any ten random number \n");

for(i=0;i<10;i++)

{

printf("a[%d]=",i+1);

scanf("%d",&a[i]);

}

for(i=0;i<10;i++)

{for(j=i;j<10;j++)

{

if(a[i]>a[j])

{

k=a[i];

a[i]=a[j];

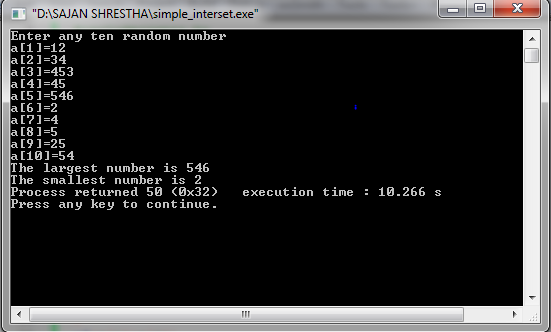
a[j]=k;

}}}

printf("The largest number is %d\nThe smallest number is %d",a[9],a[0]);

}

Output:



Title:

# WAP to find if the entered number is palindrome or not.

Source code:

#include<stdio.h>

#include<conio.h>

palindrome();

int main()

{

palindrome();

}

palindrome()

{

intn,i,r,sum=0,a,b;

printf("Enter any number\n");

scanf("%d",&n);

a=n;

while(n>0)

{

r=n%10;

sum=sum\*10+r;

n=n/10;

}

if (sum==a)

{

printf("%d is a palindrome number\n ",a);

}

else

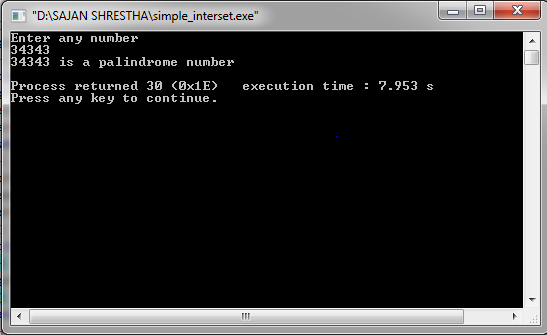
{

printf("%d is not a palindrome number \n",a);

}

}

Output:



Title:

# WAP to input the number and reverse it.

Source code:

#include<stdio.h>

#include<conio.h>

rev();

int main()

{

rev();

}

rev()

{

intn,i,r,sum=0,a,b;

printf("Enter any number\n");

scanf("%d",&n);

a=n;

while(n>0)

{

r=n%10;

sum=sum\*10+r;

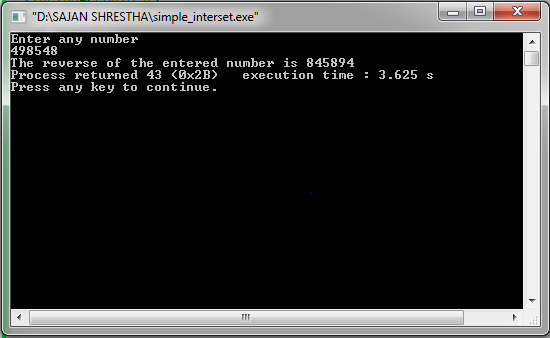
n=n/10;

}

printf("The reverse of the entered number is %d",sum);

}

Output:



Title:

# WAP to find factorial using recursive function.

Source code:

#include<stdio.h>

#include<conio.h>

intfac(int );

void main()

{

inti,j,k;

printf("Enter the number to find the factorial\n");

scanf("%d",&i);

k=fac(i);

printf("The factorial of the entered number is %d",k);

}

intfac(int x)

{

if(x==0)

{

return 1;

}

else

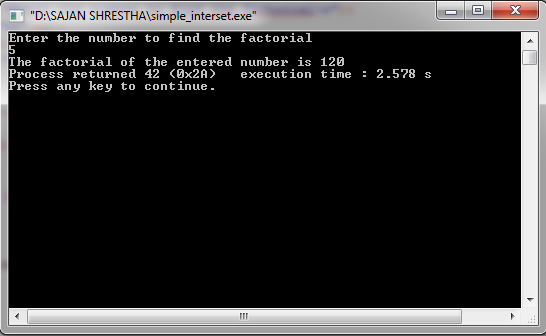
{

return(x\*fac(x-1));

}

}

Output:



**CONCLUSION**

From the above practical, I got to know about different types of functions, function prototypes and function call. Also, I became able to find out area and circumference of a circle, find out largest and smallest number, check whether the number is palindrome ,find reverse and factorial of the given number using function in C-programming language.

Title:

# WAP to read Name, Address, age and salary of N number of employee and find the average salary.

Source code:

#include<stdio.h>

#include<conio.h>

#include<string.h>

structemp

{int s;

char n[20],a[20];};

void main()

{

inti,j,nu,k=0,av;

printf("Enter the total number of employee in your company\n");

scanf("%d",&nu);

structemp e[nu];

for(i=0;i<nu;i++)

{

printf("Enter the details of %d employee\n",i+1);

printf("Name[%d]=\t",i+1);

scanf("%s",&e[i].n);

printf("Adderes=\t");

scanf("%s",&e[i].a);

printf("Salary=\t");

scanf("%d",&e[i].s);

}

for(i=0;i<nu;i++)

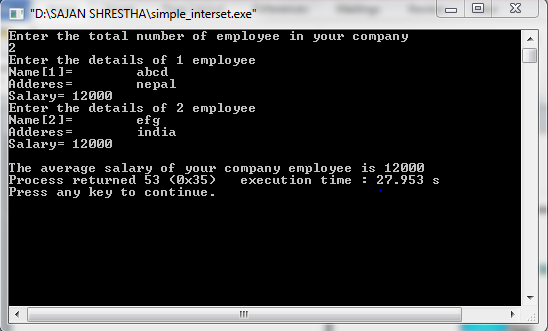
{k=k+e[i].s;}

av=k/nu;

printf("\nThe average salary of your company employee is %d",av);

}

Output:



Title:

# WAP to store the record of N customer in a bank with fields account number, name and balance and display the records of the customer who has the highest balance in the bank.

Source code:

#include<stdio.h>

#include<conio.h>

struct bank

{intac,b;

charna[30];};

void main()

{inti,j,n,k;

printf("Enter the total number of customers:");

scanf("%d",&n);

struct bank a[n];

for(i=0;i<n;i++)

{printf("Enter the details of customer %d:\n",i+1);

printf("Account number=\t");

scanf("%d",&a[i].ac);

printf("Name=\t");

scanf("%s",a[i].na);

printf("Balance=\t");

scanf("%d",&a[i].b);

printf("\n\n");}

for(i=0;i<n;i++)

{for(j=i;j<n;j++)

{if(a[i].b<a[j].b)

{k=a[i].b;

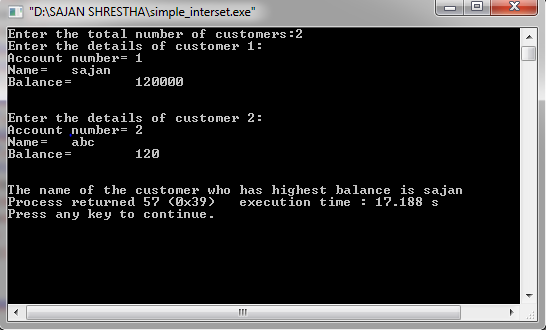
a[i].b=a[j].b;

a[j].b=k;

}}}

printf("The name of the customer who has highest balance is %s",a[0].na);}

Output:



Title:

# WAP to read Roll number, Name and marks in 6 subjects and display the result with percentage and grade( Prepare Grade sheet of one student).

Source code:

#include<stdio.h>

#include<conio.h>

#include<string.h>

structst{

introll,m[6];

char name[20];

}a;

void main()

{

inti,j,k;

float t=0,g,p,sum=0;

printf("Enter the details of one student \n");

printf("Enter the name of the student \n");

scanf("%s",a.name);

printf("Enter the roll number of the student\n");

scanf("%d",&a.roll);

for(i=0;i<6;i++){

printf("m[%d]=",i+1);

scanf("%d",&a.m[i]);

}

for(i=0;i<6;i++){

sum=sum+a.m[i];

}

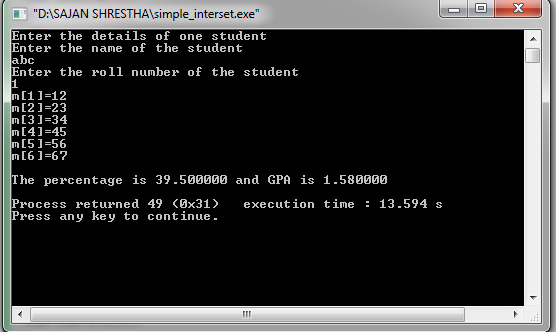
p=sum/6;

g=p/25;

printf("\nThe percentage is %f and GPA is %f\n",p,g);

}

Output:



**CONCLUSION**

After completing the lab work on structure and union, I got the information about user defined data type “structure”. Also, I got the idea about calculating average salary, finding customer with highest balance and displaying result of students with percentage and grade using structure in C-programming language.

Title:

# WAP to swap 2 numbers using pointer.

Source code:

#include<stdio.h>

#include<string.h>

void swap(int\*,int\*);

void main()

{

int \*i,\*j,a,b;

printf("Enter two numbers:");

scanf("%d%d",&a,&b);

i=&a;

j=&b;

printf("The value of character variable a and b before swapping is %d and %d respectively.\n",a,b);

swap(i,j);

printf("The value of character variable a and b after swapping is %d and %d respectively.\n ",a,b);

}

void swap(int \*i,int \*j)

{

int k;

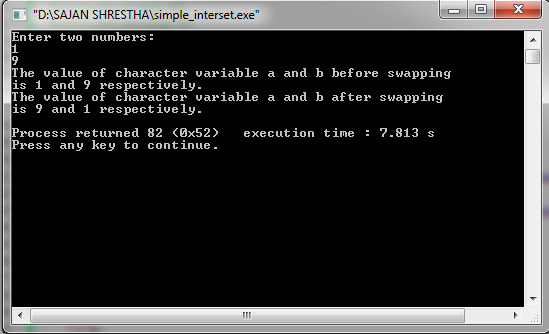
k=\*i;

\*i=\*j;

\*j=k;

}

Output:



Title:

# WAP to sort the n number given by the user.

Source code:

#include<stdio.h>

#include<conio,h>

void main()

{

intn,i,j,k;

printf("Enter the total number of number\n");

scanf("%d",&n);

int a[n];

for(i=0;i<n;i++)

{

printf("a[%d]=\t",i+1);

scanf("%d",&a[i]);

}

for(i=0;i<n;i++)

{

for(j=i;j<n;j++)

{

if(\*(a+i)>\*(a+j))

{

k=\*(a+i);

\*(a+i)=\*(a+j);

\*(a+j)=k;

}}}

printf("\n");

for(i=0;i<n;i++)

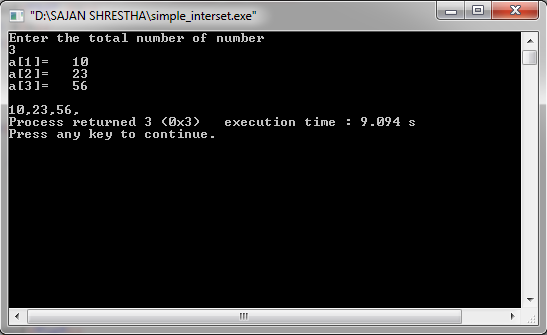
{

printf("%d,",a[i]);

}

}

Output:



**CONCLUSION**

After accomplishing the lab work on pointer, I got the idea about the uses of pointer while finding the address of particular variable. Along with this, I became able to swap two numbers and sorting the numbers in ascending or descending order using pointer in a C-programming language.

Title:

# WAP on file handling concept.

1. Create a data file named Grade12.dat and write record (roll no, name, class, percentage) until n is typed

2. Read all records from the same file and display on the screen in appropriate format

3. Display records of students who have got distinction.

4. Delete delete particular record from the data file on the basis of roll no.( ask the user for the data to be deleted)

Source code:

#include<stdio.h>

#include<string.h>

void main()

{

inti,j,k,n,roll,per,class,a,b;

char c[5]="yes",name[20],s,p;

printf("Enter 'a' to add the details of student\n");

printf("Enter 'r' to display the record\n");

printf("Enter 'p' to display the record of those student who has got distinction\n");

printf("Enter 'd' to delete the record of specific student\n");

scanf("%c",&s);

switch(s)

{

case 'a':

{

FILE \*fp;

fp=fopen("record.dat","a");

if(fp==0)

{

printf("The file is not created\n");

}

else

printf("The file is created\n");

do

{

printf("\nName=\t");

scanf("%s",name);

printf("class=\t");

scanf("%d",&class);

printf("roll no=\t");

scanf("%d",&roll);

printf("percentage=\t");

scanf("%d",&per);

fprintf(fp,"%s\t\t%d\t%d\t%d\n",name,class,roll,per);

printf("type yes to add \nelse type no \n");

scanf("%s",c);

} while (strcmp(c,"no")!=0);

fclose(fp);

}

break;

case 'r':

{

FILE \*fp;

fp=fopen("record.dat","r");

if(fp==0)

{

printf("The file is not created\n");

}

else

printf("The file is created\n");

printf("Name\t\tClass\tRoll\tpercentage\n");

while((p=fgetc(fp))!=EOF)

{

printf("%c",p);

}

fclose(fp);

}

break;

case 'p':

{

FILE \*fp;

fp=fopen("record.dat","r");

if(fp==0)

{

printf("The file is not created\n");

}

else

printf("The file is created\n");

printf("\nThe details of student who got distionction are\n");

printf("Name\t\tclass\troll\tpercentage\n");

while((fscanf(fp,"%s%d%d%d",name,&class,&roll,&per))!=EOF)

{

if(per>80)

{

printf("%s\t\t%d\t%d\t%d\n",name,class,roll,per);

}

}

fclose(fp);

}

break;

case 'd':

{

FILE \*fp,\*gp;

fp=fopen("record.dat","r");

gp=fopen("record1.dat","w");

if(fp==0&&gp==0)

{

printf("The file is not created\n");

}

else

printf("The file is created\n");

intdel\_roll;

printf("Enter the roll number of student whose details is to be deleted\n");

scanf("%d",&del\_roll);

while((fscanf(fp,"%s%d%d%d",name,&class,&roll,&per))!=EOF)

{

if(del\_roll!=roll)

{

fprintf(gp,"%s\t\t%d\t%d\t%d\n",name,class,roll,per);

}

}

fclose(fp);

fclose(gp);

FILE \*sp,\*rp;

sp=fopen("record.dat","w");

rp=fopen("record1.dat","r");

if(rp==0&&sp==0)

{

printf("The deletion is not done\n");

}

else

printf("The deletion is in process...\n");

while((fscanf(rp,"%s%d%d%d",name,&class,&roll,&per))!=EOF)

{

fprintf(sp,"%s\t\t%d\t%d\t%d\n",name,class,roll,per);

}

fclose(sp);

fclose(rp);

}

break;

default:

printf("Invalid choice.");

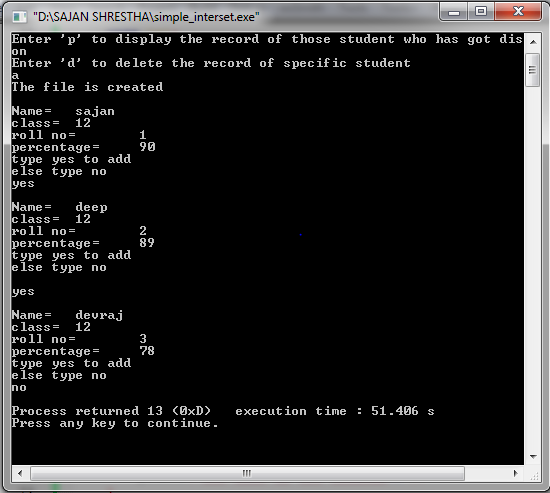
break;

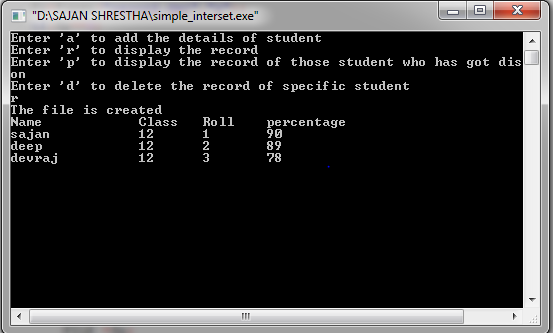
}

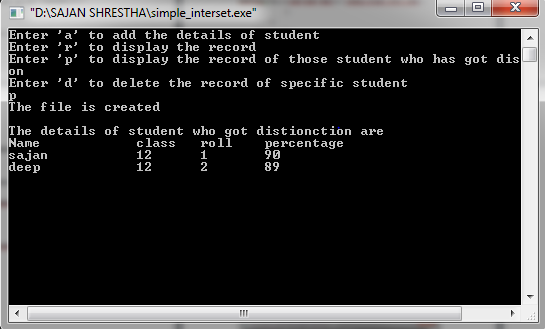
getch();

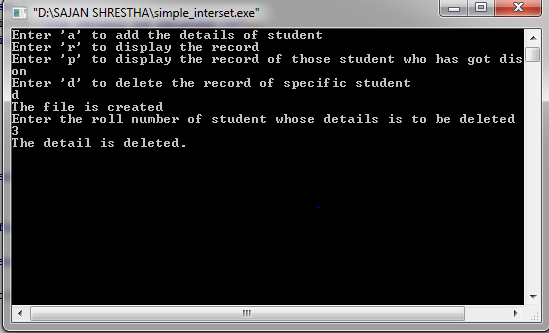
}

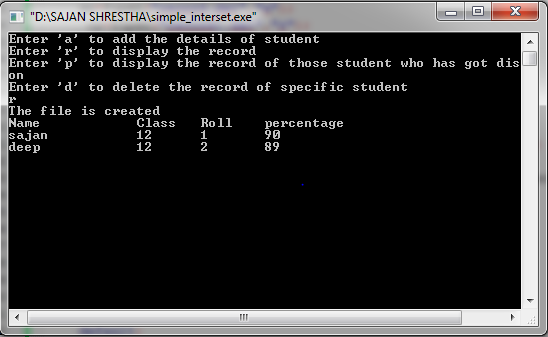
Output:











**CONCLUSION**

After completing the lab work on File handling, I gained knowledge about file handling functions and file modes (r, w, a).Also, I was able to create a file ,write the details of the students in it , read the details of the students from it, display the records of student getting distinction and deleting a particular record from the file using file handling function in a C-programming language.

**INTRODUCTION**

JavaScript often abbreviated JS, is a [programming language](https://en.wikipedia.org/wiki/Programming_language) that is one of the core technologies of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web), alongside [HTML](https://en.wikipedia.org/wiki/HTML) and [CSS](https://en.wikipedia.org/wiki/CSS). Over 97% of [websites](https://en.wikipedia.org/wiki/Website) use JavaScript on the [client](https://en.wikipedia.org/wiki/Client_(computing)) side for [web page](https://en.wikipedia.org/wiki/Web_page) behavior, often incorporating third-party [libraries](https://en.wikipedia.org/wiki/Library_(computing)). All major [web browsers](https://en.wikipedia.org/wiki/Web_browser) have a dedicated [JavaScript engine](https://en.wikipedia.org/wiki/JavaScript_engine) to execute the [code](https://en.wikipedia.org/wiki/Source_code) on [users](https://en.wikipedia.org/wiki/User_(computing))' devices.

JavaScript is a [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), often [just-in-time compiled](https://en.wikipedia.org/wiki/Just-in-time_compilation) language that conforms to the [ECMAScript](https://en.wikipedia.org/wiki/ECMAScript" \o "ECMAScript) standard. It has [dynamic typing](https://en.wikipedia.org/wiki/Dynamic_typing), [prototype-based](https://en.wikipedia.org/wiki/Prototype-based_programming) [object-orientation](https://en.wikipedia.org/wiki/Object-oriented_programming), and [first-class functions](https://en.wikipedia.org/wiki/First-class_function). It is [multi-paradigm](https://en.wikipedia.org/wiki/Programming_paradigm), supporting [event-driven](https://en.wikipedia.org/wiki/Event-driven_programming), [functional](https://en.wikipedia.org/wiki/Functional_programming), and [imperative](https://en.wikipedia.org/wiki/Imperative_programming) [programming styles](https://en.wikipedia.org/wiki/Programming_paradigm). It has [application programming interfaces](https://en.wikipedia.org/wiki/Application_programming_interface) (APIs) for working with text, dates, [regular expressions](https://en.wikipedia.org/wiki/Regular_expression), standard [data structures](https://en.wikipedia.org/wiki/Data_structure), and the [Document Object Model](https://en.wikipedia.org/wiki/Document_Object_Model) (DOM).

The ECMAScript standard does not include any [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O), such as [networking](https://en.wikipedia.org/wiki/Computer_network), [storage](https://en.wikipedia.org/wiki/Data_storage), or [graphics](https://en.wikipedia.org/wiki/Computer_graphics) facilities. In practice, the web browser or other [runtime system](https://en.wikipedia.org/wiki/Runtime_system) provides JavaScript APIs for I/O.

JavaScript engines were originally used only in web browsers, but are now core components of some [servers](https://en.wikipedia.org/wiki/Server_(computing)) and a variety of [applications](https://en.wikipedia.org/wiki/Application_software). The most popular runtime system for this usage is [Node.js](https://en.wikipedia.org/wiki/Node.js).

Although [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) and JavaScript are similar in name, [syntax](https://en.wikipedia.org/wiki/Syntax_(programming_languages)), and respective [standard libraries](https://en.wikipedia.org/wiki/Standard_library), the two languages are distinct and differ greatly in design

**Advantages of Java Script**

* Speed. Client-side JavaScript is very fast because it can be run immediately within the client-side browser. Unless outside resources are required, JavaScript is unhindered by network calls to a backend server.
* Simplicity. JavaScript is relatively simple to learn and implement.
* Popularity. JavaScript is used everywhere on the web.
* Interoperability. JavaScript plays nicely with other languages and can be used in a huge variety of applications.
* Server Load. Being client-side reduces the demand on the website server.
* Gives the ability to create rich interfaces.

**Disadvantages of Java Script**

* Client-Side Security. Because the code executes on the users’ computer, in some cases it can be exploited for malicious purposes. This is one reason some people choose to disable Javascript.
* Browser Support. JavaScript is sometimes interpreted differently by different browsers. This makes it somewhat difficult to write cross-browser code

**Features of Java Script**

* JavaScript is a object-based scripting language.
* Giving the user more control over the browser.
* It Handling dates and time.
* It Detecting the user's browser and OS,
* It is light weighted.
* JavaScript is a scripting language and it is not java.
* JavaScript is interpreter based scripting language.
* JavaScript is case sensitive.
* JavaScript is object based language as it provides predefined objects.
* Every statement in java script must be terminated with semicolon (;).
* Most of the java script control statements syntax is same as syntax of control

# OPERATORS

## Source:

<html>

<head>

<title>operators concept</title>

</head>

<body>

<script>

var a = 5,

b = 6,

c;

c = a + b;

document.write("The sum of " + a + " + " + b + " = " + c + ".");

if (a > 0 && b > 0) {

document.write("The value of both a and b is greater than 0.");

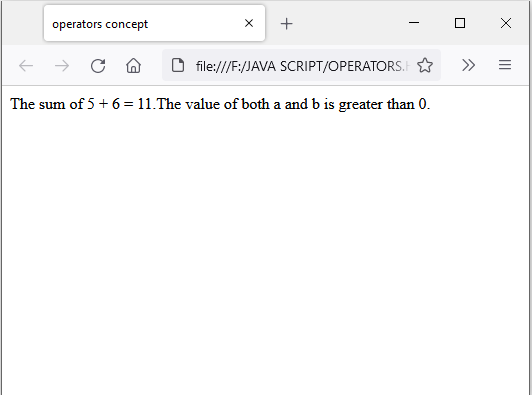
}

</script>

</body>

</html>

## Output:



# IF ELSE

## Source:

<html>

<head>

<title>IF ELSE</title>

<body>

<script>

var a = "five";

if (a >0) {

document.write(a + " is a non zero

postivenumber. ");

} elseif (a <0) {

document.write(a + " is negative number. ");

} elseif (a == 0) {

document.write(a + " is zero.");

} else {

document.write(a + " is invalid input.");

}

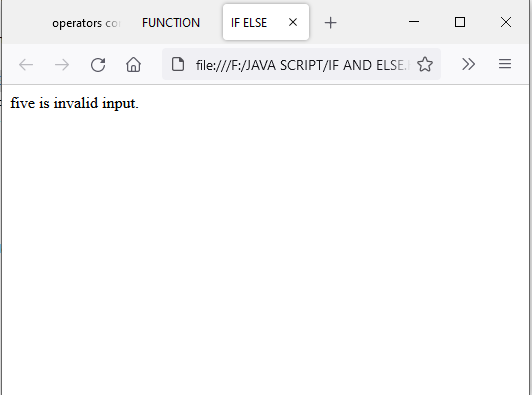
</script>

</body>

</head>

</html>

## Output:



# LOOP

## Source:

<html>

<head><title>LOOP</title></head>

<body>

<p id="p1">Finding the greatest number among the given numbers.</p><br />

<script>

var a = ["8", "0", "9", "6"];

document.write("The numbers are:");

for (i = 0; i < 4; i++) {

document.write(", " + a[i]); }

for (i = 0; i < 4; i++) {

for (j = i; j < 4; j++) {

if (a[i] > a[j]) {

k = a[i];

a[i] = a[j];

a[j] = k; } } }

</script><br />

<script>

document.write("The ascending order of given numbers are:");

for (i = 0; i < 4; i++) {

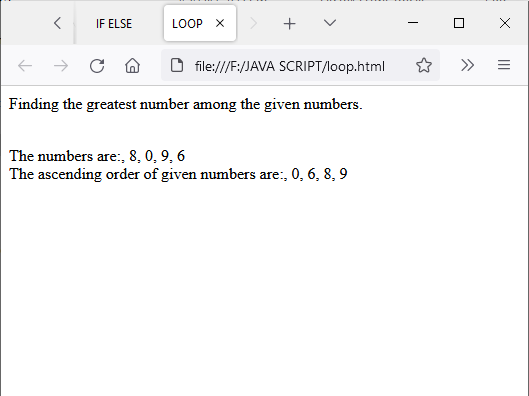
document.write(", " + a[i]); }

</script>

</body>

</html>

## Output:



# SWITCH

## Source:

<html>

<head><title>SWITCH</title></head>

<body><script>

var x = 8,

y = 9;

var c = "\*";

switch (c) {

case"+":

document.write("The sum of " + x + " and " + y + " is " + (x + y));

break;

case"-":

document.write(

"The difference between " + x + " and " + y + " is " + (x - y) );

break;

case"\*":

document.write(

"The multiplication of " + x + " and " + y + " is " + x \* y);

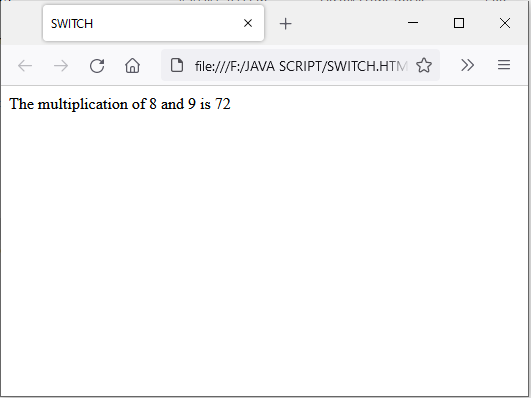
break; }

</script>

</body>

</html>

## Output:



# FUNCTION

## Source:

<html>

<head>

<title>FUNCTION</title>

</head>

<body>

<script>

function hello() {

document.write("Hello! ");

}

hello();

function sum(x, y) {

var i;

i = x + y;

return i;

}

var i;

var x = 5,

y = 3;

i = sum(x, y);

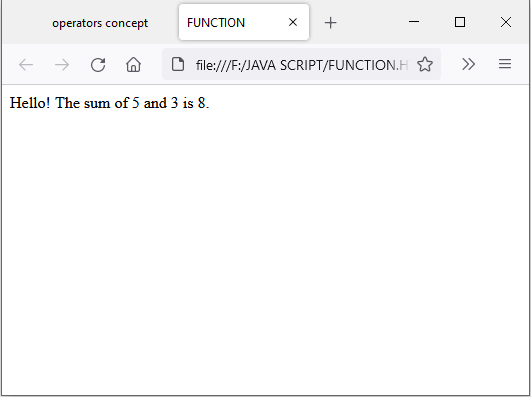
document.write("The sum of " + x + " and " + y + " is " + i + ".");

</script>

</body>

</html>

## Output:



# CLASS

## Source:

<html>

<head>

<title>

CLASS CONCEPT

</title>

</head>

<body>

<script>

class student{

constructor(name,grade,mark)

{

this.na=name;

this.g=grade;

this.m=mark;

}

}

deep= new student("deep khadka","12","100");

ashim= new student("ashim lamsal","12","200");

sajan= new student("sajan shrestha","12","100")

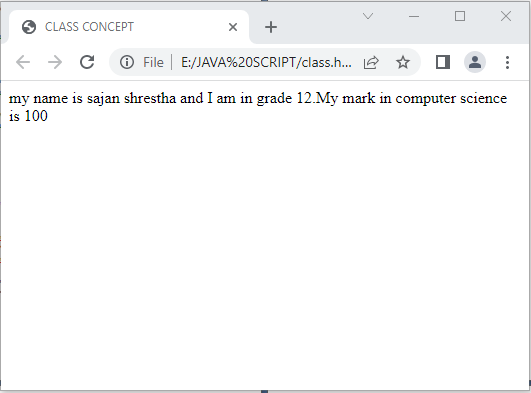
document.write("my name is "+sajan.na+" and I am in grade "+sajan.g+".My mark in computer science is "+sajan.m);

</script>

</body>

</html>

## Output:



# OBJECT

Object is a instant of class. An object is a standalone entity, with properties and type. It is an entity having state and behavior(properties and method).

## Source:

<html>

<head>

<title>

object concept

</title>

</head>

<body>

<script>

var i=5;

student=

{

name:"sajanshrestha",

rollno:"1",

faculty:"science",

address:"danchhi"

}

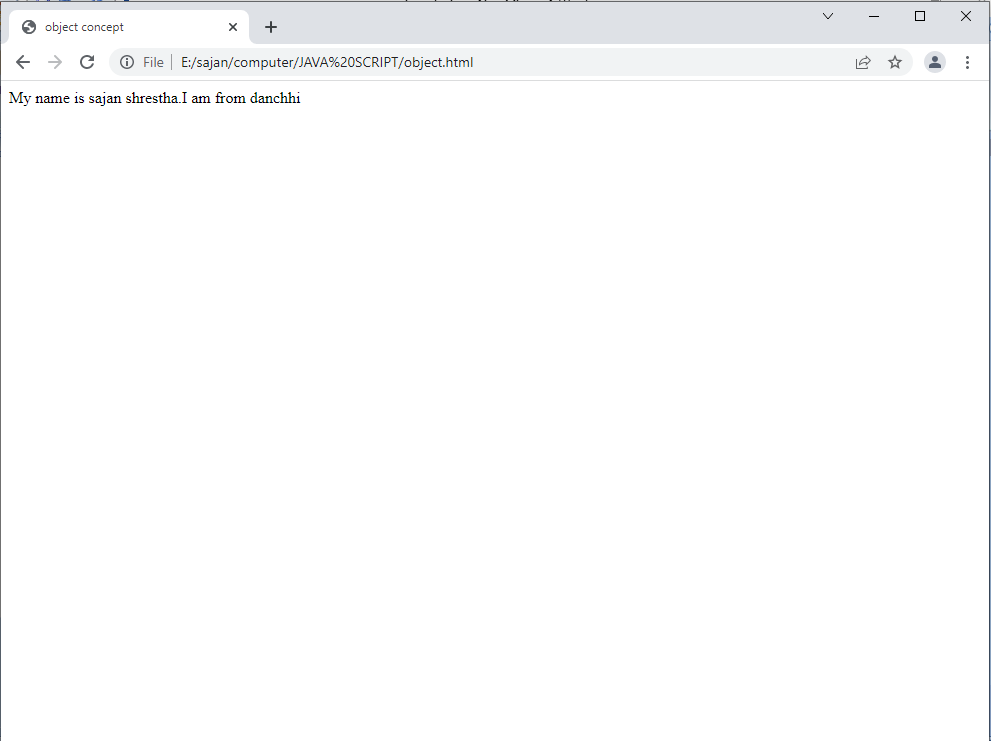
document.write("My name is "+student.name+".I am from "+student.address);

</script>

</body>

</html>

## Output



# PRE-DEFINE OBJECT

## Source:

<html>

<head>

<title>PRE DEFINED</title>

</head>

<body>

<button id="bj">Click here to find time</button>

<p id="pj"></p>

<script>

function time() {

var d = new Date();

var m = d.getMinutes();

var s = d.getSeconds();

var h = d.getHours();

document.getElementById("pj").innerHTML =

h + "Hrs, " + m + "Min and " + s + "sec";

}

var b = document.getElementById("bj");

b.onclick = time;

</script>

</body>

</html>

## Output:

# EVENT HANDLING

## Source:

<html>

<head>

<title>HTML EVENT HANDLER</title>

<body>

<script>

function myname() {

document.write("my name is sajan shrestha");

}

</script>

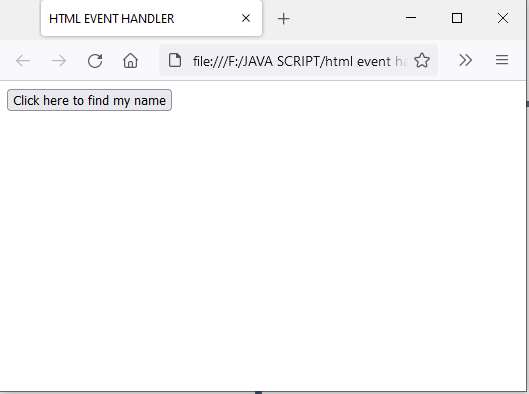
<button onclick="myname()">Click here to find my name</button>

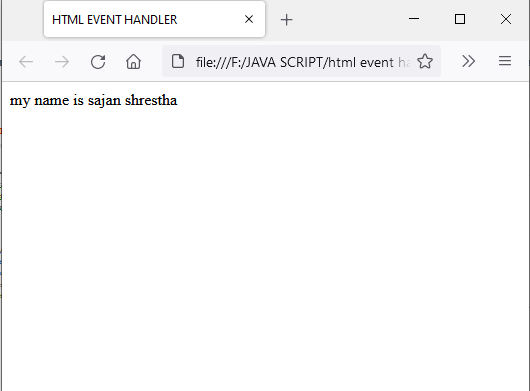
</body>

</head>

</html>

## Output:





# FORM VALIDATION

## Source:

<!DOCTYPE html>

<head>

<title>FORM VALIDATION</title>

</head>

<body>

<script>

function valid(){

var name\_valid=document.form.name.value;

var address\_valid=document.form.address.value;

if(name\_valid==address\_valid){

return true;}

else{

alert("!Incorrect password!");

return false;}}

</script>

<form name="form"  onsubmit=" return valid()"><br>

Enter new password:<br>

<input type="password" name="name"><br>

Re-Enter new password:

<br><input type="password" name="address"><br>

<input type="submit">

</form>

</body>

</html>

## Output:

s