**ASSIGNMENT 1**

Roll No: 075bct093

/\*1. Write a program to implement the following two functions on string using pointer:

i. To calculate length of given string\*/

#include<iostream>

using namespace std;

int main(){

int strsize(char \*);

char str[100];

char \*ptr;

ptr=str;

int n;

cout<<"Enter a string:"<<endl;

cin>>ptr;

n=strsize(ptr);

cout<<"Length of string is"<<n;

}

int strsize(char \*ptr)

{

int count=0;

while(\*ptr!='\0')

{

count++;

\*ptr++;

}

return (count);

}

// ii. To concatenate two string using pointer

#include<iostream>

using namespace std;

int main(){

char str1[100],str2[100];

char \*ptr1;

ptr1=str1;

char \*ptr2;

ptr2=str2;

int n,i=0,j=0;

cout<<"Enter a string:"<<endl;

cin>>str1;

cout<<"Enter next string:"<<endl;

cin>>str2;

while(str1[i]!='\0'){

++ptr1;

i++;

}

while(str2[j]!='\0'){

\*ptr1=\*ptr2;

ptr1++;

ptr2++;

j++;

}

cout<<"The string after concatenation is:"<<str1;

return 0;

}

/\*2. WAP to create a structure DATE with its member: day, month and year.

Write a function which takes two pointer variables of type DATE to calculate the age of person (i.e. pass structure by reference).

\*/

#include<iostream>

using namespace std;

struct DATE{

int day;

int month;

int year;

DATE() {}

DATE(int d, int m, int y)

{

day = d;

month = m;

year = y;

}

};

DATE\* CalculateAge(DATE\* present, DATE\* dob)

{

int month[] = { 31, 28, 31, 30, 31, 30, 31,

31, 30, 31, 30, 31 };

DATE \*temp = new DATE(0, 0, 0);

if (dob->day > present->day)

{

present->day = present->day + month[dob->month - 1];

present->month = present->month -1;

}

if(dob->month > present->month)

{

present->year = present->year - 1;

present->month = present->month + 12;

}

temp->day = present->day - dob->day;

temp->month = present->month - dob->month;

temp->year = present->year - dob->year;

return temp;

}

int main()

{

DATE \*person=new DATE;

DATE \*now = new DATE(1, 9, 2020);

cout<<"Enter date of birth"<<endl;

cin>>person->day>>person->month>>person->year;

DATE \*age = CalculateAge(now, person);

cout << age->year << " years " << age->month << " months " << age->day << " days old";

}

//3. WAP to swap the values (of type int, double, char) using function template.

#include<iostream>

using namespace std;

template<typename T>

class zwap

{

T a,b;

public:

zwap(T a1, T b1) : a(a1), b(b1){ } ;

void swap(){

T temp = a;

a = b;

b = temp;

}

void display(){

cout << a << " " << b << endl;

}

};

int main()

{

zwap<int> i(2, 3);

zwap<char> c('A', 'Z');

zwap<double> d(4.5,10.2245);

i.display();

i.swap();

i.display();

c.display();

c.swap();

c.display();

d.display();

d.swap();

d.display();

}

//4. WAP to swap the values (of type int, double, char) using class template.

#include<iostream>

using namespace std;

template<class T>

void Swap(T &a, T &b){

T temp;

temp=a;

a=b;

b=temp;

}

int main(){

int i1= 5 , i2 = 20;

char ch1 ='T', ch2 = 'P';

double d1=40.223, d2=2.03354;

cout << i1 << " " << i2 << endl;

cout <<ch1 << " " << ch2 << endl;

cout << d1 << " " << d2 << endl;

cout<<"After Swap"<<endl;

Swap(i1, i2);

Swap(ch1, ch2);

Swap(d1, d2);

cout <<i1 <<" "<< i2 << endl;

cout <<ch1 << " " << ch2 << endl;

cout <<d1 <<" "<< d2 << endl;

return 0;

}

/\*Write a C++ program to perform arithmetic operations on two numbers and throw an exception if the dividend is zero or does not contain an operator.

Enter the input as a+b, where ‘a’ and ‘b’ are input numbers and ‘+’ as operator.

Check for the valid operators and perform the different operations like addition, subtraction, multiplication and division accordingly.

Perform no.5 without using Exception Class\*/

#include<iostream>

using namespace std;

#include<iostream>

using namespace std;

int main(){

float a,b,add,sub,mul,div;

char opr;

char msg1[100];

int msg2;

cout<<"Enter the expression as- a(operation)b: "<<endl;

cin>>a>>opr>>b;

try{

if(opr=='-'){sub=a-b;cout<<sub<<endl;}

else if(opr=='+'){add=a+b;cout<<add<<endl;}

else if(opr=='\*'){mul=a\*b;cout<<mul<<endl;}

else if(opr=='/'){

if(b!=0){div=a/b;cout<<div<<endl;}

else{throw msg1;}

}

else{throw msg2;}

}

catch(char a[]){

cout<<"Denominator must be non zero.Try again"<<endl;

main();

}

catch(int ){

cout<<"Try again with valid operator.Try again"<<endl;

main();

}

return 0;}

//Perform no.5 using Exception Class

#include<stdio.h>

using namespace std;

#include<iostream>

using namespace std;

class except{

public:

except(){

cout<<"Cannot divide by zero ";

}

};

int main(){

float a,b,add,sub,mul,div;

char opr;

char msg1[100];

int msg2;

cout<<"Enter the expression as- a(operation)b: "<<endl;

cin>>a>>opr>>b;

if(opr=='-'){sub=a-b;cout<<sub<<endl;}

else if(opr=='+'){add=a+b;cout<<add<<endl;}

else if(opr=='\*'){mul=a\*b;cout<<mul<<endl;}

else if(opr=='/'){

if(b!=0){div=a/b;cout<<div<<endl;}

else{throw except();}

}

else{ cout<<"Try again with valid operator.Try again"<<endl;} }