Assignment 8

**Program 1**

/\*"Declare a template class called exam having an array of generic type as a data member,

named elements[10]. Define following generic (template) member functions:

- sort to arrange elements in ascending order

- find\_max to find and return maximum from the array

Define main to illustrate usage of these functions to process two different types of data."\*/

#include <iostream>

#include <conio.h>

using namespace std;

template <class T>

class Sort

{

private:

T n[5];

public:

void get()

{

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

{

cout<<"Enter value "<<i+1<<" : ";

cin >> n[i];

}

}

T findMax()

{

int i;

T max;

max=n[0];

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

{

if(max < n[i])

max=n[i];

}

return max;

}

void sort()

{

int i, j;

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

{

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

{

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

{

T element;

element = n[i];

n[i] = n[j];

n[j] = element;

}

}

}

}

void display()

{

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

cout<<n[i]<<"\n";

}

};

int main()

{

Sort <int>intn;

Sort <float>floatn;

//system("cls");

cout << "Integer Values \n";

intn.get();

cout << "Float values \n";

floatn.get();

//system("cls");

cout<<"Int maximum: "<<intn.findMax()<<endl;

cout <<"Int sorting:"<< endl;

intn.sort();

intn.display();

cout<<"float maximum: "<<floatn.findMax()<<endl;

cout<<"float sorting:" << endl;

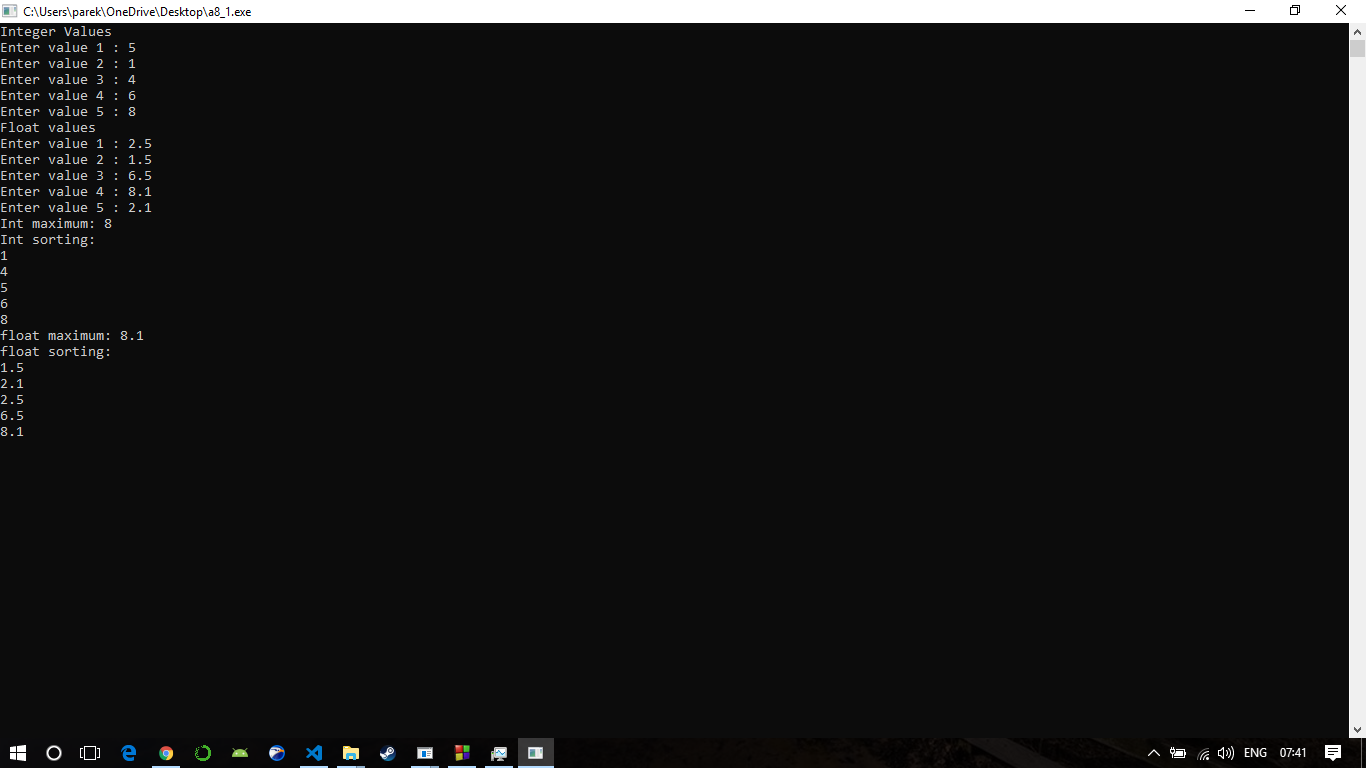
floatn.sort();

floatn.display();

getch();

return 0;

}



**Program 2**

/\*Write a function template for finding the minimum value contained in an array\*/

#include<iostream>

#include<conio.h>

using namespace std;

template <class T>

T getmin(T a[],int n)

{

int i;

T min;

min=a[0];

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

{

if(a[i]<min)

min=a[i];

}

return (min);

}

int main()

{

//system("cls");

int ia[4];

char ca[4];

float fa[4];

int i;

cout<<"Enter 4 int:";

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

cin>>ia[i];

cout<<"Enter 4 char:";

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

cin>>ca[i];

cout<<"Enter 4 float:";

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

cin>>fa[i];

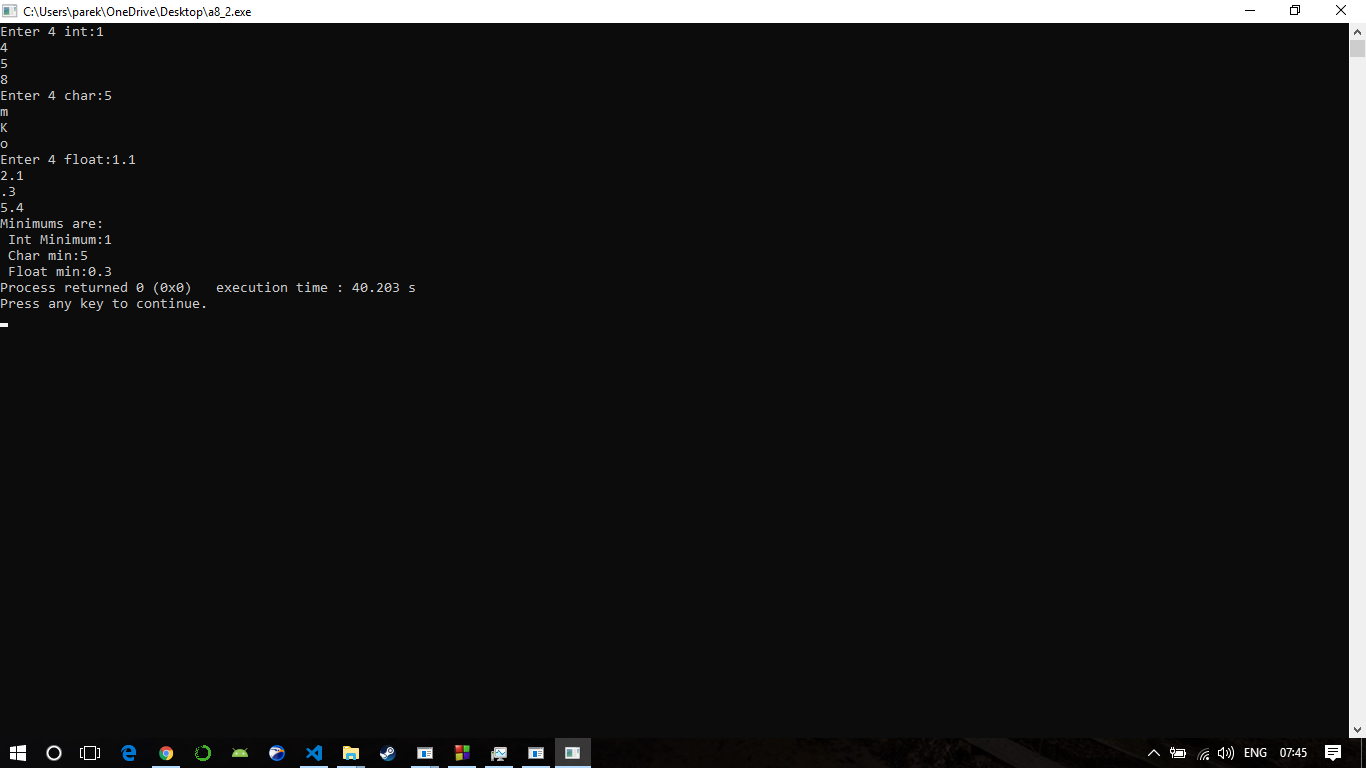
cout<<"Minimums are:";

cout<<"\n Int Minimum:"<<getmin(ia,4)<<"\n Char min:"<<getmin(ca,4)<<"\n Float min:"<<getmin(fa,4);

//getch();

return 0;

}



**Program 3**

/\*Create a generic class stack using template and implement common Push and Pop operations for different data types.\*/

#include<iostream>

#include<conio.h>

using namespace std;

template <class T>

class stack

{

private:

T s[10];

int top;

public:

stack()

{

top = -1;

}

void push(T i)

{

top++;

s[top]=i;

cout<<"\n"<<s[top]<<" is successfully pushed";

}

T pop()

{

T x=s[top];

top--;

return x;

}

void disp()

{

int i;

cout<<"\n";

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

{

cout<<s[i]<<" ";

}

}

};

int main()

{

stack <int>s1;

//system("cls");

s1.push(1);

s1.disp();

s1.push(5);

s1.disp();

s1.push(6);

s1.disp();

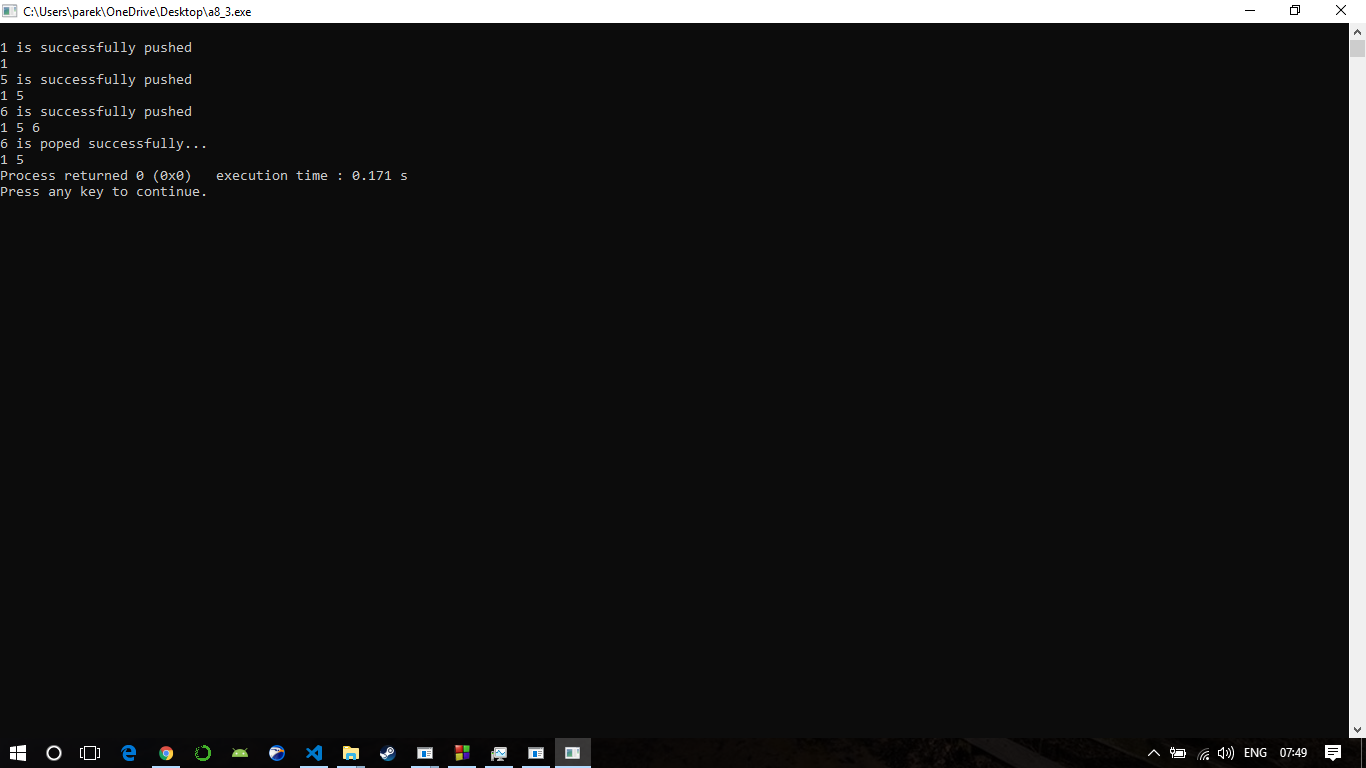
cout<<"\n"<<s1.pop()<<" is poped successfully...";

s1.disp();

//getch();

return 0;

}



**Program 4**

/\*WAP that illustrates the application of multiple catch statements\*/

#include<iostream>

using namespace std;

void test(int x)

{

try

{

if(x>0)

throw x;

else

throw 'x';

}

catch(int x)

{

cout<<"Catch an integer and that integer is:"<<x<<endl;

}

catch(char x)

{

cout<<"Catch a character and that character is:"<<x;

}

}

int main()

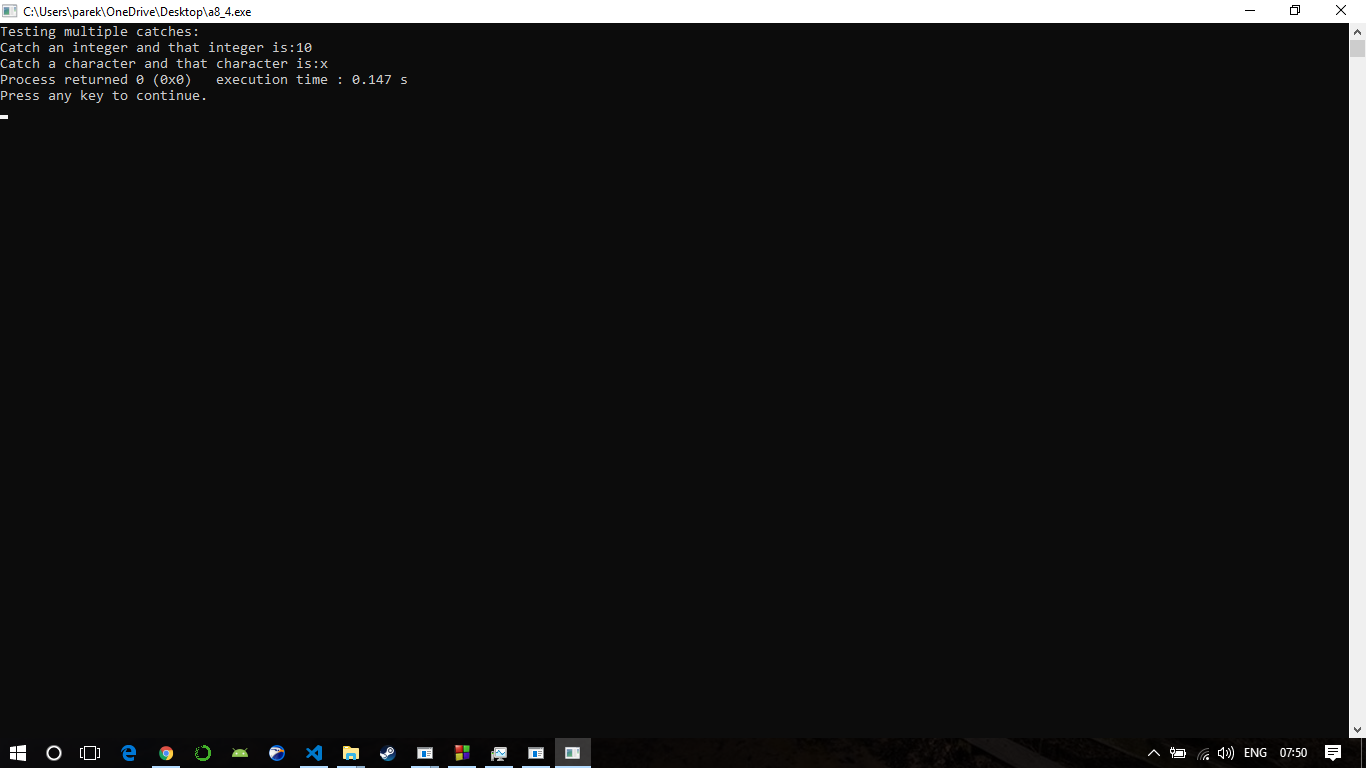
{

cout<<"Testing multiple catches:\n";

test(10);

test(0);

return 0;}



**Program 5**

/\*W.A.P. that throws an arithmetic exception whenever the result of arithmetic computation

becomes divisible by 3.\*/

#include<iostream>

#include<conio.h>

using namespace std;

int main()

{

int x=12,y=6;

//system("cls");

try

{

if((x-y)%3==0)

throw x-y;

}

catch(int z)

{

cout<<"caught:"<<z;

}

//getch();

return 0;

}

