# TASK 1:

#include <iostream>

using namespace std;

int Div(int num, int den)

{

if (den == 0)

{

throw runtime\_error("Cannot Divide by Zero !");

}

return (num / den);

}

int main()

{

int a, b, result;

cout << "Enter Numenator: ";

cin >> a;

cout << "Enter denominator: ";

cin >> b;

try

{

result = Div(a, b);

cout << "The Answer is " << result << endl;

}

catch (runtime\_error& e)

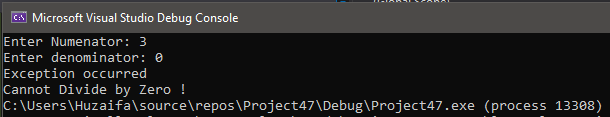
{

cout << "Exception occurred" << endl << e.what();

}

# }

# OUTPUT:



# TASK 1(b): #include<iostream>

using namespace std;

int main()

{

float a = 5.6;

try

{

throw 0;

}

catch (int i)

{

cout << "Exception thrown: " << i << endl;

}

try

{

throw a;

}

catch (float f)

{

cout << "Exception thrown: " << f << endl;

}

try

{

throw "char";

}

catch (const char\* s)

{

cout << "Exception thrown: " << s << endl;

}

try

{

throw 'q';

}

catch (char c)

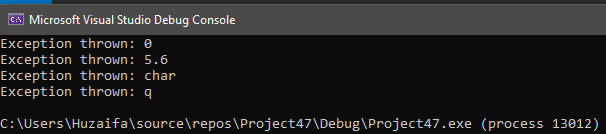
{

cout << "Exception thrown: " << c << endl;

}

# }

# OUTPUT:



# TASK 2:

#include<iostream>

using namespace std;

void addFractions(int num1, int den1, int num2, int den2, int result\_num, int result\_den)

{

try {

cout << "First fraction: ";

cout << "Enter Numenator: ";

cin >> num1;

cout << "Enter Denominator: ";

cin >> den1;

throw invalid\_argument("Entered Number is not Valid");

cout << "Second fraction: ";

cout << "Enter Numenator: ";

cin >> num2;

cout << "Enter Denominator: ";

cin >> den2;

throw invalid\_argument("Entered Number is not Valid");

result\_num = num1 \* den2 + num2 \* den1;

result\_den = den1 \* den2;

cout << "The sum of the two fractions is " << result\_num << "/" << result\_den;

}

catch (exception eObj)

{

cout << "Exception: " << eObj.what() << endl;

}

}

void SubFraction(int num1, int den1, int num2, int den2, int result\_num, int result\_den)

{

try {

cout << "First fraction: ";

cout << "Enter Numenator: ";

cin >> num1;

cout << "Enter Denominator: ";

cin >> den1;

throw invalid\_argument("Invalid Input");

cout << "Second fraction: ";

cout << "Enter Numenator: ";

cin >> num2;

cout << "Enter Denominator: ";

cin >> den2;

throw invalid\_argument("Invalid Input");

result\_num = num1 \* den2 - num2 \* den1;

result\_den = den1 \* den2;

cout << "To subtract the equation we get: " << result\_num << "/" << result\_den;

}

catch (exception eObj)

{

cout << "Exception: " << eObj.what() << endl;

}

}

void multiplyFractions(int num1, int den1, int num2, int den2, int result\_num, int result\_den)

{

cout << "First fraction: ";

cout << "Enter Numenator: ";

cin >> num1;

cout << "Enter Denominator: ";

cin >> den1;

cout << "Second fraction: ";

cout << "Enter Numenator: ";

cin >> num2;

cout << "Enter Denominator: ";

cin >> den2;

result\_num = num1 \* num2;

result\_den = den1 \* den2;

cout << "To subtract the equation we get: " << result\_num << "/" << result\_den;

}

void divideFractions(int num1, int den1, int num2, int den2, int result\_num, int result\_den)

{

cout << "First fraction: ";

cout << "Enter Numenator: ";

cin >> num1;

cout << "Enter Denominator: ";

cin >> den1;

cout << "Second fraction: ";

cout << "Enter Numenator: ";

cin >> num2;

cout << "Enter Denominator: ";

cin >> den2;

try {

if (den1 == 0 || den2 == 0)

{

throw exception("Cannot divide by zero.");

}

result\_num = num1 \* den2;

cout << result\_num << endl;

result\_den = den1 \* num2;

cout << result\_den << endl;

cout << "Division of two Fraction we get: " << result\_num << "/" << result\_den << endl;

}

catch (exception eObj)

{

cout << "Exception: " << eObj.what() << endl;

}

}

void multFractions(int num1, int den1, int num2, int den2, int result\_num, int result\_den)

{

try

{

cout << "First fraction: ";

cout << "Enter Numenator: ";

cin >> num1;

cout << "Enter Denominator: ";

cin >> den1;

throw invalid\_argument("Entered Number is not valid");

cout << "Second fraction: ";

cout << "Enter Numenator: ";

cin >> num2;

cout << "Enter Denominator: ";

cin >> den2;

throw invalid\_argument("Entered Number is not valid");

result\_num = num1 \* num2;

result\_den = den1 \* den2;

cout << "Multiplication of two Fraction we get: " << result\_num << "/" << result\_den;

}

catch (invalid\_argument& obj)

{

cout << obj.what() << endl;

}

}

void display()

{

cout << "Enter what which operations do you want to perform" << endl;

cout << " 1) Addition" << endl;

cout << " 2) Subtraction" << endl;

cout << " 3) Division" << endl;

cout << " 4) Multiplication" << endl;

}

int main()

{

int num1 = 0, num2 = 0, den1 = 0, den2 = 0;

int result\_num = 0, result\_den = 0;

int choice;

display();

cout << "Enter Choice: ";

cin >> choice;

if (choice == 1)

{

addFractions(num1, den1, num2, den2, result\_num, result\_den);

}

else if (choice == 2)

{

SubFraction(num1, den1, num2, den2, result\_num, result\_den);

}

else if (choice == 3)

{

divideFractions(num1, den1, num2, den2, result\_num, result\_den);

}

else if (choice == 4)

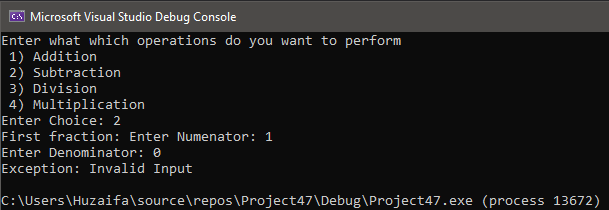
{

multFractions(num1, den1, num2, den2, result\_num, result\_den);

}

}

# OUTPUT:



# TASK 3:

#include <iostream>

using namespace std;

int main()

{

int lowerLimit = 50;

try

{

cout << "Entering the try block." << endl;

if (lowerLimit < 100)

{

throw exception("Less than Lower limit");

}

cout << "Exiting the try block." << endl;

}

catch (exception eObj)

{

cout << "Exception: " << eObj.what() << endl;

}

cout << "After the catch block" << endl;

}

# OUTPUT:

# TASK 4:

#include <iostream>

using namespace std;

int main()

{

double feet, inches, centimeter;

bool invalidInput = true;

while (invalidInput)

{

try

{

cout << "Enter a length in feet: ";

cin >> feet;

cout << "Enter a length in inches: ";

cin >> inches;

if (feet < 0 || inches < 0)

{

throw invalid\_argument("Negative value exception");

}

invalidInput = false;

}

catch (invalid\_argument& obj)

{

cout << obj.what() << endl;

}

}

centimeter = (feet \* 12) \* (2.54 + inches) \* 2.54;

cout << "The length in centimeters is " << centimeter << endl;

}

# OUTPUT:

# TASK 5:

#include<iostream>

using namespace std;

class invalidHr

{

public:

invalidHr()

{

cout << "Invalid Hour are entered" << endl;

}

};

class invalidMin

{

public:

invalidMin()

{

cout << "Invalid Minutes are entered" << endl;

}

};

class invalidSec

{

public:

invalidSec()

{

cout << "Invalid seconds are entered" << endl;

}

};

int main()

{

int hr, min, sec;

cout << "Enter hr: ";

cin >> hr;

cout << "Enter min: ";

cin >> min;

cout << "Enter sec: ";

cin >> sec;

try

{

if (hr < 0 || hr>12)

{

throw invalidHr();

}

else if (min < 0 || min>60)

{

throw invalidMin();

}

else if (sec < 0 || sec >60)

{

throw invalidSec();

}

else {

cout << "Time is " << hr + 12 << ":" << min << ":" << sec << endl;

}

}

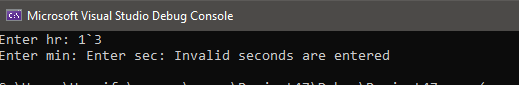
catch (invalidHr h) {}

catch (invalidMin m) {}

catch (invalidSec s) {}

}

# OUTPUT:



# Task 6:

#include<iostream>

using namespace std;

class invalidDay

{

public:

invalidDay()

{

cout << "Invalid Day is entered" << endl;

}

};

class invalidMonth

{

public:

invalidMonth()

{

cout << "Invalid Month is entered" << endl;

}

};

class invalidyear

{

public:

invalidyear()

{

cout << "Invalid year is entered" << endl;

}

};

int main()

{

int day, month, year;

cout << "Enter Birth Day: ";

cin >> day;

cout << "Enter Birth Month: ";

cin >> month;

cout << "Enter Birth Year: ";

cin >> year;

try

{

if (day < 0 || day>31)

{

throw invalidDay();

}

else if (month < 0 || month>12)

{

throw invalidMonth();

}

else if (year < 0)

{

throw invalidyear();

}

else

{

cout << "Date of Birth is " << day << "/" << month << "/" << year << endl;

}

}

catch (invalidDay d) {}

catch (invalidMonth m) {}

catch (invalidyear y) {}

}

# Output:

