**Test program**

#include <iostream>

using namespace std;

class Test

{

  int marks[5];

public:

  void setMarks()

  {

    cout << "Enter Marks for 5 Subjects: ";

    for (int i = 0; i < 5; i++)

    {

      cin >> marks[i];

    }

  }

  int operator[](int i)

  {

    if(i < 0 || i >= 5)

    {

      cout << "Invalid Index" << endl;

      exit(0);

    }

    return marks[i];

  }

  void compare(Test t)

  {

    for (int i = 0; i < 5; i++)

    {

      if (marks[i] > t.marks[i])

      {

        cout << "Marks of Subject " << i + 1 << " are more in Object 1" << endl;

      }

      else if (marks[i] < t.marks[i])

      {

        cout << "Marks of Subject " << i + 1 << " are more in Object 2" << endl;

      }

      else

      {

        cout << "Marks of Subject " << i + 1 << " are equal in both Objects" << endl;

      }

    }

  }

};

int main()

{

  Test t1, t2;

  t1.setMarks();

  t2.setMarks();

  t1.compare(t2);

  return 0;

}

**Triangle**

#include <iostream>

#include <cmath>

#include <stdexcept>

using namespace std;

class TriangleException

{

public:

  TriangleException()

  {

    cout << "Exception: Sum of Two Sides is Less than Third Side" << endl;

  }

};

class Triangle

{

  float a, b, c;

public:

  void setSides()

  {

    cout << "Enter Three Sides of Triangle: ";

    cin >> a >> b >> c;

  }

  void calculateArea()

  {

    if (a + b <= c || b + c <= a || c + a <= b)

    {

      throw TriangleException();

    }

    float s = (a + b + c) / 2;

    float area = sqrt(s \* (s - a) \* (s - b) \* (s - c));

    cout << "Area of Triangle: " << area << endl;

  }

};

int main()

{

  Triangle t;

  t.setSides();

  try

  {

    t.calculateArea();

  }

  catch (TriangleException e)

  {

  }

  catch (...)

  {

    cout << "Exception occurred!!" << endl;

  }

  return 0;

}

**(ANY TWO: Thinking Cap & Product GST)**

**Thinking Cap Exception**

#include <iostream>

#include <string.h>

#include <stdexcept>

using namespace std;

class thinkingCap

{

  char green[10];

  char red[10];

public:

  void slots(char newGreen[], char newRed[])

  {

    if (strlen(newGreen) > 10 || strlen(newRed) > 10)

    {

      throw "Exception: String length is greater than 10";

    }

    strcpy(green, newGreen);

    strcpy(red, newRed);

  }

  void pushGreen() *const*

  {

    cout << "Green: " << green << endl;

  }

  void pushRed() *const*

  {

    cout << "Red: " << red << endl;

  }

};

int main()

{

  thinkingCap user1, user2;

  try

  {

    user1.slots("Green", "Red");

    user1.pushGreen();

    user1.pushRed();

    user2.slots("Greengreengreen", "Red");

    user2.pushGreen();

    user2.pushRed();

  }

  catch (*const* char \*msg)

  {

    cout << msg << endl;

  }

  catch (...)

  {

    cout << "Exception occurred!!" << endl;

  }

}

**Product GST with Exception**

#include <iostream>

#include <string>

#include <stdexcept>

using namespace std;

class GST;

class Product

{

  string product\_name;

  float product\_price;

public:

  void setProduct()

  {

    cout << "Enter Product Name: ";

    cin >> product\_name;

    cout << "Enter Product Price: ";

    cin >> product\_price;

  }

*friend* void finalPrice(Product p, GST g);

};

class GST

{

  float gst\_rate;

public:

  void setGST()

  {

    cout << "Enter GST Rate: ";

    cin >> gst\_rate;

  }

*friend* void finalPrice(Product p, GST g);

};

void finalPrice(Product p, GST g)

{

  if (p.product\_price < 0)

  {

    throw "Exception: Price is Negative";

  }

  if (g.gst\_rate < 0)

  {

    throw "Exception: GST Rate is Negative";

  }

  float total\_price = p.product\_price + (p.product\_price \* g.gst\_rate / 100);

  cout << "Product Name: " << p.product\_name << endl;

  cout << "Total Price: " << total\_price << endl;

}

int main()

{

  Product p;

  GST g;

  try

  {

    p.setProduct();

    g.setGST();

    finalPrice(p, g);

  }

  catch (*const* char \*msg)

  {

    cout << msg << endl;

  }

  catch (...)

  {

    cout << "Exception occurred!!" << endl;

  }

  return 0;

}

**Thinking Cap Java**

*class* Thinking {

*private* String greenString;

*private* String redString;

*public* void setString(String greenString, String redString) {

    if (greenString.length() > 10 || redString.length() > 10) {

      System.out.println("String length should be less than 10");

      return;

    }

    this.greenString = greenString;

    this.redString = redString;

  }

*public* void pushGreen() {

    System.out.println("Green: " + greenString);

  }

*public* void pushRed() {

    System.out.println("Red: " + redString);

  }

}

*public* *class* ThinkingCap {

*public* *static* void main(String[] args) {

    Thinking cap = new Thinking();

    cap.setString("Green", "Red");

    cap.pushGreen();

    cap.pushRed();

  }

}

**Lottery (Optional)**

import *java.util.Scanner*;

*public* *class* Lottery {

*public* *static* void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int lottery = (int) (Math.random() \* 100);

        System.out.print("Enter your lottery pick (two digits): ");

        int guess = input.nextInt();

        int lotteryDigit1 = lottery / 10;

        int lotteryDigit2 = lottery % 10;

        int guessDigit1 = guess / 10;

        int guessDigit2 = guess % 10;

        System.out.println("The lottery number is " + lottery);

        if (guess == lottery) {

            System.out.println("Exact match: you win Rs.10000");

        } else if (guessDigit2 == lotteryDigit1 && guessDigit1 == lotteryDigit2) {

            System.out.println("Match all digits: you win Rs.3000");

        } else if (guessDigit1 == lotteryDigit1 || guessDigit1 == lotteryDigit2 || guessDigit2 == lotteryDigit1 || guessDigit2 == lotteryDigit2) {

            System.out.println("Match one digit: you win Rs.1000");

          } else {

            System.out.println("Sorry, no match");

          }

        input.close();

    }

}