

PPL Assignment 3

Name: Himani Verma

Admission No: U19CS075

1. Write a code that performs username validation for a website. When the username is too short it should throw an exception such that it prints Too short: n (where n is the length of the given username). The final program should print Valid (if the username is valid), Invalid (if the username is invalid), or Too short: n (where n is the length of the too-short username). Make necessary assumptions if required.

Source Code:

In the below code a username is invalid only if any two consecutive characters match. It will be short length if it is of length less than 5, for all other cases it will be valid.

```
#include<string>
#include <iostream>
#include <sstream>
#include <exception>
using namespace std;

class shortlengthexception{
private:
    int n;
public:
    shortlengthexception(int x){
        n= x;
    }
    int gett() {
        return n;
    }
};

bool checkusername(string username){
    int i;
    bool check = true;
    int l = username.length();
    if(l<5){
        throw shortlengthexception(1);
    }
    for(i=0; i<l-1; i++){
        if(username[i]==username[i+1]){
            check=false;
        }
    }
    return check;
}

int WinMain(){
    string username;
    cout<<"\nEnter the Username to check validity: ";
    cin>>username;
    try{
        bool check = checkusername(username);
```

```

        if(check){
            cout<<"\nValid";
        }
        else{
            cout<<"\nInvalid";
        }
    }
    catch(shortlengthexception n){
        cout<<"\nShort Length: "<<n.gett()<<'\\n';
    }
    return 0;
}

```

Output:

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS C:\Users\Himani\Desktop\PPL\Assignment 3> cd "c:\Users\H

Enter the Username to check validity: ornamental

Valid
PS C:\Users\Himani\Desktop\PPL\Assignment 3> cd "c:\Users\H
if ($?) { .\1 }

Enter the Username to check validity: yes

Short Length: 3
PS C:\Users\Himani\Desktop\PPL\Assignment 3> cd "c:\Users\H
if ($?) { .\1 }

Enter the Username to check validity: ornamental

Invalid
PS C:\Users\Himani\Desktop\PPL\Assignment 3> █

```

2. You are required to handle error messages while working with a small computational server that performs complex calculations. It has a function that takes 2 large numbers as its input and returns a numeric result. Unfortunately, there are various exceptions that may occur during execution. Write a program so that it prints appropriate error messages. The expected behaviour is defined as follows:
 - If the compute function runs fine with the given arguments, then print the result of the function call.
 - If it fails to allocate the memory that it needs, print Not enough memory.
 - If any other standard C++ exception occurs, print Exception: S where S is the exception's error message.
 - If any non-standard exception occurs, print Other Exceptions.

Source Code:

```

#include <iostream>
#include <exception>

```

```

#include <string>
#include <stdexcept>
#include <vector>
#include <cmath>
using namespace std;

class Server{
public:
    static int compute(long long A, long long B){
        if(A<0 || B<0){
            throw std::invalid_argument("A or B is negative");
        }
        int real=-3, complex= sqrt(-1);
        if(B==0) throw 0;
        real = (A/B)*real;
        int ans = real * A + (2*A+B)- real;
        return ans;
    }
};

int WinMain(){
    long long A,B;
    cout<<"Enter the values of A and B respectively: ";
    cin>>A>>B;
    try{
        cout<< Server::compute(A,B)<< endl;
    }
    catch(std:: bad_alloc &error){
        //Not enough memory
        cout<<"\nNot enough Space"<<endl;
    }
    catch(std:: exception &error){
        //Any standard exception
        cout<<"Exception: "<<error.what()<<endl;
    }
    catch(...){
        cout<<"Other exception"<<endl;
    }
    return 0;
}

```

Output:

```

PS C:\Users\Himani\Desktop\PPL\Assignment 3> cd "c:\User
if ($?) { .\2 }
Enter the values of A and B respectively: 2 6
10
PS C:\Users\Himani\Desktop\PPL\Assignment 3> cd "c:\User
if ($?) { .\2 }
Enter the values of A and B respectively: 5 0
Other exception
PS C:\Users\Himani\Desktop\PPL\Assignment 3>

```

3. Create a class Polar that represents the points on the plane as polar coordinates (radius and angles). Create an overloaded + operator for addition of two Polar quantities. "Adding" two points on the plane can be accomplished by adding their X coordinates and then adding their Y coordinates. This gives the X and Y coordinates of the "answer." Thus you'll need to convert two sets of polar coordinates to rectangular coordinates, add them, then convert the resulting rectangular representation back to polar. You need to use the following trigonometric formulae:

$$\begin{aligned}x &= r \cdot \cos(a); \\ y &= r \cdot \sin(a); \\ a &= \text{atan}(y/x); \text{ // arc tangent} \\ r &= \sqrt{x^2 + y^2};\end{aligned}$$

Source Code:

```
#include <iostream>
#include <math.h>
using namespace std;
#define RESET "\033[0m"

class Polar
{
public:
    float R, A, X, Y;
    void getPolarData()
    {
        cout << "\nEnter Polar coordinates: ";
        cin >> R >> A;
    }
    Polar operator+(Polar p1)
    {
        cout << "\nP1 -> ";
        cout << "\nPolar Coordinates: " << p1.R << " , " << p1.A;
        p1.A = p1.A * 3.14 / 180; // degree to radians for calculation
        // Calculating rectangular coordinates for P1
        p1.X = p1.R * cos(p1.A);
        p1.Y = p1.R * sin(p1.A);
        cout << "\nRectangular Coordinates: " << p1.X << " , " << p1.Y;

        cout << "\n\nP2 -> ";
        cout << "\nPolar Coordinates: " << R << " , " << A;
        A = A * 3.14 / 180; // degree to radians for calculation
        // Calculating rectangular coordinates for P1
        X = R * cos(A);
        Y = R * sin(A);
        cout << "\nRectangular Coordinates: " << X << " , " << Y;

        Polar res;
        // Adding rectangular coordinates to get result
        res.X = p1.X + X;
        res.Y = p1.Y + Y;
        cout << "\n\nResult -> ";
        cout << "\nX -> " << p1.X << " + " << X << " = " << res.X;
        cout << "\nY -> " << p1.Y << " + " << Y << " = " << res.Y;

        // Converting rectangular coordinates to polar
        res.A = atan(res.Y / res.X);
        res.A = res.A * 180 / 3.14; // radians to degree
    }
};
```

```

        res.R = sqrt(res.X * res.X + res.Y * res.Y);
        return res;
    }
};

int main()
{
    Polar p1, p2, res;
    p1.getPolarData();
    p2.getPolarData();
    res = p1+p2;
    cout << "\n\n( " << p1.R << " , " << p1.A << " ) + ( " << p2.R << " , " << p2.A << " )
= "
        << "( " << res.R << " , " << res.A << " )" << endl;
    return 0;
}

```

Output:

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS C:\Users\Himani\Desktop\PPL\Assignment 3> cd "c:\Users\Himani\Desktop\PPL\Assignment 3"
.\3 }

Enter Polar coordinates: 5 27.56

Enter Polar coordinates: 1 169

P1 ->
Polar Coordinates: 1 , 169
Rectangular Coordinates: -0.981341 , 0.192277

P2 ->
Polar Coordinates: 5 , 27.56
Rectangular Coordinates: 4.4332 , 2.31231

Result ->
X -> -0.981341 + 4.4332 = 3.45186
Y -> 0.192277 + 2.31231 = 2.50458

( 5 , 0.480769 ) + ( 1 , 169 ) = ( 4.26477 , 35.9819 )
PS C:\Users\Himani\Desktop\PPL\Assignment 3> 

```

4. A file contains a list of telephone numbers in the following form:

John 23456

Ken 9846

The names contain only one word and the names and telephone numbers are separated by white spaces. Write a program to read a file and display its contents in two columns. The names should be left justified and the number right justified.

Source Code:

```
#include <iostream>
#include <iomanip> //for setw
#include <string>
#include <fstream>
using namespace std;
int main()
{
    ifstream ifs("telephone.txt");

    string name;
    int val;
    while (ifs >> name >> val) //ifs reading data as datatype of string i.e name and val
    (integer)
    {
        cout << name << setw(12) << val << endl;
    }
}
```

Output:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Himani\Desktop\PPL\Assignment 3> cd "c:\Users\H
.\4 }
John      23456
Ken       9846
Lisa      4456
Jaya      2234
Esha      2837
Riya      5666
PS C:\Users\Himani\Desktop\PPL\Assignment 3> █
```