

PRINCIPLES OF PROGRAMMING LANGUAGES**ASSIGNMENT 3**

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.

CODE=>

```
//U19CS009
//Brijesh Rohit

#include<bits/stdc++.h>
using namespace std;

class users {
    int count;
    vector<string> unames;
public:
    users()
    {
        count = 0;
    }

    bool checkuser(string uname)
    {
        auto isUser = find(this->unames.begin(), this->unames.end(), uname);
        if (isUser == unames.end()) return true;
        return false;
    }

    int verify_username(string uname)
    {
        //MIN LENGTH IS 5
        if (uname.length() < 5)
        {
            return -1;
        }
        if (!isalpha(uname[0]) && uname[0] != '_')
        {
            return 0;
        }
        for (int i = 1; i < uname.length(); i++)
        {
            if (!isalnum(uname[i]) && uname[i] != '_' && uname[i] != '.')
            {
                return 0;
            }
        }
        if (!checkuser(uname))
```

```

        {
            return 0;
        }
        return 1;
    }

void adduser(string uname)
{
    if (this->verify_username(uname) == 0 || this->verify_username(uname) ==
-1)
    {
        return;
    }
    this->count++;
    this->unames.push_back(uname);
}

};

int main()
{
    users u;

    string s[5] = {"brijesh", "birju", "B@1234", "Bw_12__231"};
    for (int i = 0; i < 5; i++)
        u.adduser(s[i]);

    //USER INPUT
    string s1;
    cout << "Enter an username : ";
    cin >> s1;
    if (u.verify_username(s1) == -1)
    {
        cout << "Too Short : " << s1.length() << endl;
    }
    else if (u.verify_username(s1) == 0)
    {
        cout << "Invalid" << endl;
    }
    else
    {
        cout << "Valid" << endl;
    }
    return 0;
}

```

OUTPUT=>

```
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ g++ ppl-assign03-q1.cpp
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ ./a.out
Enter an username : __birjesh
Valid
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ ./a.out
Enter an username : bir
Too Short : 3
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ ./a.out
Enter an username : 123
Too Short : 3
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ ./a.out
Enter an username : 1123123
Invalid
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ ./a.out
Enter an username : 1232Brijesh
Invalid
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ ./a.out
Enter an username : Bij123
Valid
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ ./a.out
Enter an username : _____
Valid
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ ./a.out
Enter an username : -Beroi@
Invalid
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ ./a.out
Enter an username : _briwr@
Valid
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ ./a.out
Enter an username : brijesh.rohit
Valid
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ ./a.out
Enter an username : _brijesh_rohit
Valid
```

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 behavior 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.

CODE=>

```
//U19CS009
//Brijesh Rohit

#include <bits/stdc++.h>
using namespace std;

int complexCalc(int a, int b)
{
    int c = pow(a, b);
    int prod = 1;
    for (int i = 0; i < a + b; i++) prod *= a;
    return prod / b;
}

int main()
{
    int a, b;
    cout << "Enter value a and b : ";
    cin >> a >> b;

    try {
        int n = complexCalc(a, b);
        cout << "The result of complexCalc() : " << n << endl;
    }
    //catching bad alloc
    catch (bad_alloc & b) {
        cerr << "Bad memory allocation : " << b.what() << endl;
    }
    //catching std exception
    catch (exception & e)
    {
        cerr << "Standard Exception Caught : " << e.what() << endl;
    }
    //catching other exception
    catch (...)
    {
        cerr << "Other Exception" << endl;
    }
    return 0;
}
```

OUTPUT=>

```
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ g++ ppl-assign03-q2.cpp
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ ./a.out
Enter Value a and b : -100 19
Bad memory allocation : std::bad_array_new_length
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ ./a.out
Enter Value a and b : 1321323 1223
The result of complexCalc() : -442552
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ ./a.out
Enter Value a and b : 0 0
Floating point exception (core dumped)
```

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:

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

CODE=>

```
//U19CS009
//Brijesh Rohit

#include <bits/stdc++.h>
using namespace std;

const float pi = 3.14159; //value of pi

class Polar {
    float radius;    //in unit length
    float angle;     //in radians
public:

    Polar (float r, float th)
    {
        this->radius = r;
        this->angle = th;
    }

    void setRadius(float r )
    {
        this->radius = r;
    }
    float getRadius()
    {
        return this->radius;
    }
    void setAngle(float t )
    {
        this->angle = t;
    }
}
```

```

}
float getTheta()
{
    return this->angle;
}

pair<float, float> ConvToRect()
{
    float x = (this->radius) * cos(this->angle);
    float y = (this->radius) * sin(this->angle);
    return make_pair(x, y);
}
//operator overloading
Polar operator + (Polar obj2)
{
    pair<float, float> p1 = this->ConvToRect();
    pair<float, float> p2 = obj2.ConvToRect();
    float x, y;
    x = p1.first + p2.first;
    y = p1.second + p2.second;
    Polar P(0, 0);
    float r, th;
    r = sqrt(x * x + y * y);

    th = float(atan2(y, x));

    P.setRadius(r);
    P.setAngle(th);
    return P;
}
void display()
{
    cout << "Radius : " << this->getRadius();
    cout << "\tAngle : " << this->getTheta();
}
};

int main()
{
    float r, th;

    cout << "Enter first Polar coordinates (r,th)): ";
    cin >> r >> th;
    Polar p1(r, th);

    cout << "Enter second Polar coordinate (r,th)): ";
    cin >> r >> th;
    Polar p2(r, th);

    Polar p3(0, 0);
    cout << "The three polar coordinates are :\n";
    cout << "p1 : ";
    p1.display();

```

```

    cout << "\n";
    cout << "p2 : ";
    p2.display();
    cout << "\n";
    cout << "p3 : ";
    p3.display();
    cout << "\n";

    cout << "p3 is changed to sum of p1 and p2.\n";
    p3 = p1 + p2;
    cout << "Addition of both coordinates : ";
    p3.display();
    cout << "\n";
    return 0;
}

```

OUTPUT=>

```

brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ g++ ppl-assign03-q3.cpp
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ ./a.out
Enter first Polar coordinates (r,th)): 1 2.1
Enter second Polar coordinate (r,th)): 4 2.3
The three polar coordinates are :
p1 : Radius : 1 Angle : 2.1
p2 : Radius : 4 Angle : 2.3
p3 : Radius : 0 Angle : 0
p3 is changed to sum of p1 and p2.
Addition of both coordinates : Radius : 4.98403 Angle : 2.26013

```

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.

CODE=>

```
//U19CS009
//Brijesh Rohit

#include<bits/stdc++.h>
using namespace std;

void display()
{
    ifstream in;
    in.open("file.txt");
    string s;
    while (in)
    {
        //fetching line in s.
        getline(in, s);
        int i;
        int flag = 0;
        string temp, temp1;
        //finding first word
        for ( i = 0; i < s.length(); i++)
        {
            if (s[i] == ' ')
            {
                break;
            }
            temp += s[i];
        }
        temp[i] = '\0';
        //left align
        cout << setw(15) << left << temp;
        //finding phone number
        for (int j = i; j < s.length(); j++)
        {
            temp1 += s[j];
        }
        //right align
        cout << setw(15) << right << temp1 << endl;
    }
}

int main()
{
    display();
    return 0;
}
```

OUTPUT=>

1.file.txt

```
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ ls
a.out      ppl-assign03-q1.cpp  ppl-assign03-q3.cpp  q1.png  q3.png  u19cs009-p
pl-assign03.docx
file.txt   ppl-assign03-q2.cpp  ppl-assign03-q4.cpp  q2.png  q4.png
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ cat file.txt
Brijesh 31321
Rohit 23214
Kwinal 12431
John 23456
Neha 42143
Ken 9846
Anjali 32653
Suraj 43252
Chandan 43212
Krutik 32121
Himanshu 31244
```

2.terminal output

```
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ g++ ppl-assign03
-q4.cpp
brijesh@brijesh-GF75-Thin-9SCSR:~/Documents/ppl/ppl-assign03$ ./a.out
Brijesh          31321
Rohit            23214
Kwinal           12431
John             23456
Neha             42143
Ken              9846
Anjali           32653
Suraj            43252
Chandan          43212
Krutik           32121
Himanshu         31244
```