Q1. Write a C++ program to demonstrate the use of try, catch block with the argument as an

integer and string using multiple catch blocks.

#include <iostream>

using namespace std;

int main() {

int a;

cout << "Enter number : ";

cin >> a;

try {

if ( (a % 2) == 0)

throw a;

else

throw "odd number : ";

} catch (int a) {

cout << "even number : " << a;

} catch (const char \*s ) {

cout << s << a;

}

}

Q2. Write a C++ program to demonstrate try, throw and catch statements.

#include <iostream>

using namespace std;

int main() {

float a, b;

cout << "Enter two numbers : ";

cin >> a >> b;

try {

if (b == 0)

throw "divide by zero error";

cout << "Answer : " << a / b;

} catch (const char \*s) {

cout << s;

}

}

Q3. Write a C++ program to perform arithmetic operations on two numbers and throw an

exception if the dividend is zero or does not contain an operator.

#include <iostream>

using namespace std;

float add(int x, int y) {

return x + y;

}

float sub(int x, int y) {

return x - y;

}

float multi(int x, int y) {

return x \* y;

}

float divide(int x, int y) {

if (y == 0)

throw "Divide by zero error";

return x / y;

}

int main() {

char opr;

float a, b;

cout << "Enter 1st number : ";

cin >> a;

cout << "Enter 2nd number : ";

cin >> b;

fflush(stdin);

try {

cout << "Enter Operator : ";

cin >> opr;

if ( (opr != '+' ) && (opr != '-') && (opr != '\*') && (opr != '/') ) {

throw "does not contain an operator";

}

switch (opr) {

case '+':

cout << "Addition : " << add(a, b);

break;

case '-':

cout << "Subtraction : " << sub(a, b);

break;

case '\*':

cout << "Multiplication : " << multi(a, b);

break;

case '/':

cout << "Division : " << divide(a, b);

break;

}

} catch (const char \*s) {

cout << s;

}

}

Q4. Write a C++ program to accept an email address and throw an exception if it does not contain @ symbol.

#include <iostream>

#include <conio.h>

using namespace std;

int validEmail(char \*);

int main() {

char mail[50];

cout << "Enter email : ";

gets(mail);

try {

if (validEmail(mail)) {

throw "Not valid email";

}

} catch (const char \*s) {

cout << s;

}

}

int validEmail(char \*mail) {

for (int i = 0; mail[i] != '\0'; i++) {

if (mail[i] == '@')

return 0;

}

return 1;

}

Q5. Write a C++ program to accept a mobile number and throw an exception if it does not contain 10 digits.

#include <iostream>

using namespace std;

int main() {

char num[20];

cout << "Enter Number : ";

gets(num);

try {

if (strlen(num) > 10 || strlen(num) < 10)

throw " does not contain 10 digits";

} catch (const char \*s) {

cout << s;

}

}

Q6. Write a C++ program to accept area pin code and throw an exception if it does not contain 6 digits.

#include <iostream>

#include <string.h>

using namespace std;

int main() {

char pin[20];

cout << "Enter Pin code : ";

gets(pin);

try {

if (strlen(pin) > 6 || strlen(pin) < 6)

throw " does not contain 6 digits";

} catch (const char \*s) {

cout << s;

}

}

Q7. Write a C++ program to accept a username if the username has less than 6 characters

or does contain any digit or special symbol.

#include <iostream>

#include <string>

using namespace std;

int digit(string s);

int symbol(string s);

int main() {

string str;

cout << "Enter Username : ";

getline(cin, str);

try {

if (str.length() > 6) {

if (!digit(str)) {

if (symbol(str))

throw "if your username is greater than 6 character then include digit or special symbol";

}

}

} catch (const char \*s) {

cout << s;

}

}

int digit(string s) {

for (int i = 0; i < s.length(); i++) {

if ( isdigit(s[i]) )

return 1;

}

return 0;

}

int symbol(string s) {

for (int i = 0; i < s.length(); i++) {

if (((s[i] >= 32 && s[i] <= 47) || (s[i] >= 58 && s[i] <= 64) || (s[i] >= 91 && s[i] <= 96))) {

return 0;

}

}

return 1;

}

Q8. Write a C++ program to accept a password and throw an exception if the password has

less than 6 characters or does not contain a digit or does not contain any special

character or does not contain any capital letter.

#include <iostream>

#include <string>

using namespace std;

int digit(string s);

int symbol(string s);

int capital(string s);

int main() {

string str;

cout << "Enter Password : ";

getline(cin, str);

try {

if (str.length() < 6) {

throw "your password is lower than 6 character";

}

if (!digit(str)) {

throw "your password does not contain digit";

}

if (symbol(str)) {

throw "your password does not contain special symbol";

}

if (capital(str)) {

throw "your password does not contain capital letter";

}

} catch (const char \*s) {

cout << s;

}

}

int digit(string s) {

for (int i = 0; i < s.length(); i++) {

if ( isdigit(s[i]) )

return 1;

}

return 0;

}

int symbol(string s) {

for (int i = 0; i < s.length(); i++) {

if (((s[i] >= 32 && s[i] <= 47) || (s[i] >= 58 && s[i] <= 64) || (s[i] >= 91 && s[i] <= 96))) {

return 0;

}

}

return 1;

}

int capital(string s) {

for (int i = 0; i < s.length(); i++) {

if (s[i] >= 65 && s[i] <= 90) {

return 0;

}

}

return 1;

}

Q9.Write a C++ program to accept Gmail id only and throw an exception if the id does not

contain @ and gmail.com.

#include <iostream>

using namespace std;

int valid(string str);

int main() {

string str;

cout << "Enter Gmail id : ";

getline(cin, str);

try {

if (valid(str)) {

throw "Enter correct gmail id";

} else {

cout << "Gmail id correct";

}

} catch (const char \*s) {

cout << s;

}

}

int valid(string str) {

if (str.at(0) != '@') {

string s = "@gmail.com";

int f = str.find(s);

if (f == -1) {

return 1;

} else {

return 0;

}

} else {

return 1;

}

}

Q10. Write a C++ program to accept Nickname and throw an exception if it has greater than 8

characters or does contain a digit or special symbol or space.

#include <iostream>

#include <string>

using namespace std;

int digit(string s);

int symbol(string s);

int space(string s);

int main() {

string str;

cout << "Enter Nickname : ";

getline(cin, str);

try {

if (str.length() >= 9) {

throw "your nickname is greater than 8 character";

}

if (digit(str)) {

throw "your nickname contain digit";

}

if (symbol(str)) {

throw "your nickname contain special symbol";

}

if (space(str)) {

throw "your nickname contain space";

}

} catch (const char \*s) {

cout << s;

}

}

int digit(string s) {

for (int i = 0; i < s.length(); i++) {

if ( isdigit(s[i]) )

return 1;

}

return 0;

}

int symbol(string s) {

for (int i = 0; i < s.length(); i++) {

if (((s[i] >= 33 && s[i] <= 47) || (s[i] >= 58 && s[i] <= 64) || (s[i] >= 91 && s[i] <= 96))) {

return 1;

}

}

return 0;

}

int space(string s) {

for (int i = 0; i < s.length(); i++) {

if (s[i] == ' ') {

return 1;

}

}

return 0;

}