Course: OOPS **Session:** Exceptional Handling **Timestamp:** 2021-4-1 10:51:38 **Register Number:** RA2031241010094

Q. User defined Exception - Division

Create an custom - user defined exception for Division of two numbers.

1. Include the header file "exception".

2. Create a class "Divide_By_Zero_Exception" that inherits "exception" base class.

3. Declare the what() method and create custom exception as follows:
public:
const char * what() const throw() {
return "Divide By Zero Exception"; }

4. In main method, input the two numbers and if the second value is "ZERO" then throw the exception.
Hint: Divide_By_Zero_Exception d;
throw d;

Kindly use same object name for the class "Divide_By_Zero_Exception" as "d"

5. Use what() method to display the message and the object name for exception class should be "e".
Hint: catch(exception& e) {
cout << e.what(); }

Refer sample input and Output

Use exceptional handling keywords try and catch for develop this program. Otherwise you wont get evaluated.

Programming Language need to be used:C++

Source Code

```
#include<iostream>
#include<exception>
using namespace std;
class Divide_By_Zero_Exception:public exception
{
public:
const char *what()const throw()
{
return "Divide By Zero Exception";
}
};
int main()
{
try
{
int a,b;
cin>>a>>b;
if(b==0)
{
Divide_By_Zero_Exception d;
throw d;
}
else
{
cout<<a/b;
}
}
catch(exception& e)
{
 cout<<e.what();
    }
 return 0;
}
```

Sample Input

```
10
0
```

Sample Output

```
Divide By Zero Exception
```

Result

Thus, Program " **User defined Exception - Division** " has been successfully executed

Course: OOPS **Session:** Exceptional Handling **Timestamp:** 2021-4-1 10:52:02 **Register Number:** RA2031241010094

Q. Compare two string

Ravi is given the two string and ask the student to compare and find exception for given strings with necessary exception handling functions
Mandatory:
Use the keyword try, catch and throw.
Refer Testcase input and output.

Source Code

```
#include <iostream>
#include<string.h>
using namespace std;
int main()
{
  int d=1,j=0,i;
  char a[100],b[100];
  cin>>a;
  cin>>b;
  try
  {
    if(cin)
  {
    if(strcmp(a,b)==0)
      cout<<a<<" is "<<b;
    else
      cout<<a<<" is not "<<b;
    }
    else
      throw(d);
  }
  catch(int e)
  {
    cout<<"Invalid input Try again";
  }
 return 0;
}
```

Sample Input

```
srm
sr
```

Sample Output

```
srm is not sr
```

Result

Thus, Program " **Compare two string** " has been successfully executed

Q. Number Exception

Maths teacher is given the task to student that. Write a program to input a number num and run a loop from 0 to num. The program should throw an exception whenever the loop counter variable is a multiple of 4, and display the number of exceptions at the end. If it is not an integer then give "Invalid input".
Input:
Enter the number of iterations : 12
Output:
Number of exceptions :3

Mandatory:

Use the keyword try, catch and throw.

Refer Testcase input and output.

Source Code

```
#include <iostream>
using namespace std;
int main()
{
  int a,b;
  cin>>a;
  try
  {
    if(cin)
    {
    if(a>0)
    {
      b=a/4;
      cout<<"Number of exceptions: "<<b;
      }
    else
    {
      cout<<"Invalid input";
    }
    }
    else
    throw(a);
  }
  catch(int e)
  {
     cout<<"Invalid input";
  }
 return 0;
}
```

Sample Input

12

Sample Output

Number of exceptions: 3

Result

Thus, Program " **Number Exception** " has been successfully executed

Q. Reverse - Array Exceptions

Bogar, the Tamil (Mother of all languages) Siddhar was given a task for checking the Arrays Size in C++.

Agathiyar, another siddhar was given the opportunity to select the "ARRAY OPERATIONS" to be assigned for Bogar.

Agathiyar after consulting with 16 Siddhars in Lemuria Continent called as "Kumari Kandam" and decided to assigned "Reversing the Array Operations" to Bogar. Now Bogar wants to check the Array Size and find reverse of the array.

Bogar was asked to implement the concept using Exceptional Handling (try, catch and throw)

Throw exception of the array size is negative or greater than 20

Refer Sample Input and Output

Programming Language need to be used:C++

Source Code

```
#include <iostream>
using namespace std;
int main()
{
  int a,b[1000],i;
  cin>>a;
  try
  {
    if(a<=20&&a>0)
    {
      for(i=0;i<a;i++)
        cin>>b[i];
      for(i=a-1;i>=0;i--)
        cout<<b[i]<<" ";
    }
    else
    {
      throw(a);
    }
  }
  catch(int e)
  {
 cout<<"Exception occurred";
  }
 return 0;
}
```

Sample Input

```
14
12 5 4 3 2 1 0 9 8 7 6 5 4 3
```

Sample Output

```
3 4 5 6 7 8 9 0 1 2 3 4 5 12
```

Result

Thus, Program " **Reverse - Array Exceptions** " has been successfully executed

Course: OOPS **Session:** Exceptional Handling **Timestamp:** 2021-4-1 10:53:16 **Register Number:** RA2031241010094

Q. Checking Valid Data

Bogar was given a task to check whether the entered mark is valid or not. Bogar framed three rules for checking the validity of the mark
Rule 1: The mark should be greater than 0 and less than or equal to 100 [0 < m <=100]
Rule 2: The mark should not exceed 100.
Rule 3: No negative Marks
Rule 4: It should be a valid integer number
Kindly help Bogar - the Tamil SIDDHAR to perform the operations using exceptional handling.
Mandatory:
Use exceptional handling keywords try and catch for develop this program. Otherwise you wont get evaluated

Source Code

```
#include <iostream>
#include<math.h>
using namespace std;
int main()
{
  int a;
  cin>>a;
  try
  {
    if(cin)
    {
    if(a>0&&a<=100)
    {
      cout<<"Valid Mark";
    }
    else
    {
      cout<<"Invalid Mark";
    }
    }
    else
    {
  throw(a);
    }
  }
    catch(int e)
    {
 cout<<"Invalid input. try again";
    }
 return 0;
}
```

Sample Input

125

Sample Output

Invalid Mark

Result

Thus, Program " **Checking Valid Data** " has been successfully executed

Course: OOPS **Session:** Exceptional Handling **Timestamp:** 2021-4-1 10:54:23 **Register Number:** RA2031241010094

Q. Palindrome

Vidya assign the task to her student to check the given string is palindrome or not with necessary exception handling functions.
Input : Albhabets only allowed.
Mandatory:
Use the keyword try, catch and throw.
Refer Testcase input and output.

Source Code

```
#include <iostream>
#include<string.h>
using namespace std;
int main()
{
  int d=1,j=0,i;
  char a[100],b[100];
  cin>>a;
  for(i=strlen(a)-1;i>=0;i--)
  {
    b[j]=a[i];
    j++;
  }
  try
  {
    if(strcmp(a,b)==0)
      cout<<a<<" is a palindrome";
    else
      throw(d);
  }
  catch(int e)
  {
    cout<<a<<" is not a palindrome";
  }
 return 0;
}
```

Sample Input

madam

Sample Output

madam is a palindrome

Result

Thus, Program " **Palindrome** " has been successfully executed

Course: OOPS **Session:** Exceptional Handling **Timestamp:** 2021-4-1 10:54:48 **Register Number:** RA2031241010094

Q. Factorial

Raman assign the task to his student to calculate the factorial of the given number with necessary exception handling functions

Mandatory:

Use the keyword try, catch and throw.

Refer Testcase input and output.

Source Code

```
#include <iostream>
using namespace std;
int main()
{
  int n,i,f=1;
  cin>>n;
  try
  {
    if(n>0&&n<20)
    {
      for(i=1;i<n+1;i++)
        f=f*i;
      cout<<"Factorial of a given Number is= "<<f;
    }
    else if(n<0)
      throw(0);
      if(n>20)
        throw(1.0);
  }
  catch(int e)
  {
 cout<<"Factorial of a given Number is= ",i;
  }
 return 0;
}
```

Sample Input

4

Sample Output

Factorial of a given Number is= 24

Result

Thus, Program " **Factorial** " has been successfully executed

Q. Check input

Vidya mam given the task to student to find the given number is integer or float
If it is not a number then it should print Invalid input
Mandatory:
Use the keyword try, catch and throw.
Refer Testcase input and output.

Source Code

```
#include <iostream>
using namespace std;
int main()
{
  int nr=0;
  char ch;
  try
  {
    cin>>nr;
    if(cin)
    {
      cin.get(ch);
       if(ch=='.')
       {
         cout<<"Floting"<<endl;
       }
      else
      {
        cout<<"Integer"<<endl;
      }
    }
    else
    {
      throw nr;
    }
  }
  catch(int e)
  {
    cout<<"Invalid input";
  }
 return 0;
}
```

Sample Input

3

Sample Output

Integer

Result

Thus, Program " **Check input** " has been successfully executed

Course: OOPS **Session:** Exceptional Handling **Timestamp:** 2021-4-1 10:55:40 **Register Number:** RA2031241010094

Q. Vowels - Consonants Exceptional Handling

Siva the faculty of SRM has given the task of validating the given input,where the students needs to get the string and find the number of vowels, consonants and white spaces.
No numeric values are allowed in the input.If the input has the numeric values then the numeric exception needs to be thrown.
Mandatory:
You should use "catch" and "throw" concepts.
Refer sample input and output.
Programming Language need to be used:C++

Source Code

```
#include <iostream>
#include<string.h>
using namespace std;
int main()
{
  char a[30];
  int len,i,vow=0,cons=0;
  cin>>a;
  len=strlen(a);
  for(i=0;i<len;i++)
  {
    try
    {
      if((a[i]>=65&&a[i]<=90)||(a[i]>=97&&a[i]<=122))
      {
        if(a[i]=='a'||a[i]=='e'||a[i]=='i'||a[i]=='o'||a[i]=='u'||a[i]=='A'||a[i]=='E'||a[i]=='I'||a[i]=='O'||a[i]=='U')
        {
          vow++;
        }
        else
        {
          cons++;
        }
      }
      else
      {
        throw a[i];
      }
    }
    catch(char c)
    {
      cout<<"Exception Caught Numeric Value";
 return 0;
    }
  }
    cout<<"Vowels="<<vow;
    cout<<endl<<"Consonants="<<cons;
    return 0;
}
```

Sample Input

CAREUniversityBUILDINGSRM

Sample Output

Vowels=9
Consonants=16

Result

Thus, Program " **Vowels - Consonants Exceptional Handling** " has been successfully executed

Q. Multiple Exception - Default Exception

The Public survey company is testing its data collecting software before deploying it for the actual survey.For that purpose you have to create a logic which give particular message or throws an exception according to the input received.

Mandaroty:

1. Get the integer input from the user (From 1 to N)
2. If the input is "1" the throw "Integer" exception and print the output as "Integer Exception" and value as "25"
3. If the input is "2" the throw "float" exception and print the output as "Float Exception" and value as "25.23"
4. If the input is greater than zero then throw default exception

Hint: catch(...)
Output: Default Exception

Explanation:

The program will throw an exception after you input something. If the number is a 1 then an integer is thrown. If the input is a 2 then a float is thrown. If it is neither of these two (not an integer or float) the default exception handler is used. This default exception handler uses the ellipsis () as the parameter of catch.

The handler will catch any exception no matter what the type of the throw exception is. (In this case a string is used.)

Refer sample input and Output

Programming Language need to be used:C++

Source Code

```
#include <iostream>
using namespace std;
int main()
{
  int n;
  cin>>n;
  try
  {
    if(n==1)
      cout<<"Integer Exception\nException number=25";
    else if(n==2)
      cout<<"Float Exception\nException number=25.23";
    else
      throw(n);
  }
  catch(...)
  {
    cout<<"Default Exception\nWrong Number Used, Input 1 or 2";
  }
 return 0;
}
```

Sample Input

```
1
```

Sample Output

```
Integer Exception
Exception number=25
```

Result

Thus, Program " **Multiple Exception - Default Exception** " has been successfully executed