object-oriented programming

(OOP)

KIAN _ ACADEMY

Exception Handling

- **C++ Exceptions:**

- When executing C++ code, different errors can occur: coding errors made by the programmer, errors due to wrong input, or other unforeseeable things.

- To Solve This Problem we use try And catch

Exception Handling

- **C++ try and catch:**
- Exception handling in C++ consist of three keywords: try, throw and catch:
  - The try statement allows you to define a block of code to be tested for errors while it is being executed.
  - The throw keyword throws an exception when a problem is detected,
  - The catch statement allows you to define a block of code to be executed, if an error occurs in the try block.

Example :

```
#include<iostream>
#include<algorithm>
#include<string>
#include <cstring>
#include<bits/stdc++.h>
using namespace std;
int main()
{
    try
    {
        int age = 15;
        if (age >= 18)
          {
cout << "Access granted - you are old enough."<<endl;
    }
     else
    {
     throw (age);//Integer object
     }
    }
catch (int myNum)
    {
        cout << "Access denied - You must be at least 18 years old"<<endl;
        cout << "Age is: " << myNum<<endl;;
    }
    return 0;
}
```

Another Example

```
#include<iostream>
#include<algorithm>
#include<string>
#include <cstring>
#include<bits/stdc++.h>
using namespace std;
int main()
{
 try
   {
      int age = 15;
      if (age >= 18)
      {
          cout << "Access granted - you are old enough.";
      }
else
      {
          throw 505;
      }
   }
   catch (int myNum)
   {
      cout << "Access denied - You must be at least 18 years old.\n";
      cout << "Error number: " << myNum;
   }
   return 0;
}
```

Another Example

```
#include<iostream>
#include<algorithm>
#include<string>
#include <cstring>
#include<bits/stdc++.h>
using namespace std;
int main()
{
int x = 10, y = 0;
    try
    {
        if (y == 0)
            throw "division by zero
Exception";
        else
            cout << x / y << endl;
    }
 catch (const char* msg)
    {
        cout << msg <<endl;
        cout << "Y must be greater than
0" << endl;
    }
    cout << "The Program Continued"
<< endl;
    return 0;
}
```

Another Example

```
#include<iostream>
#include<algorithm>
#include<string>
#include <cstring>
#include<bits/stdc++.h>
using namespace std;
int main()
{
int x = 10, y = 0;
  try
  {
    if (y == 0)
      throw exception();
    else
      cout << x / y << endl;
  }
catch (exception e)
  {
    cout << e.what()<<endl;
  }
  cout << "The Program Continued"
<< endl;
  return 0;
}
```

- **Handle Any Type of Exceptions (...)**

If you do not know the throw type used in the try block, you can use the "three dots" syntax (...) inside the catch block, which will handle any type of exception:

Example

```
#include<iostream>
#include<algorithm>
#include<string>
#include <cstring>
#include<bits/stdc++.h>
using namespace std;
int main()
{
try
   {
      int age = 15;
      if (age >= 18)
      {
cout << "Access granted - you are old enough."<<endl;
      }
      else if(age==16)
      {
         throw 16;
      }
      else
      {
         throw "karim";//habdaya to understand
      }
   }
catch (...)
   {
      cout << "Access denied - You must be at least 18 years old.\n";
   }
   cout << "The Program Continued" << endl;
   return 0;
}
```

Exam Questions:

- 1. What is an exception in C++ program?

A) A problem that arises during the execution of a program

B) A problem that arises during compilation

C) Also known as the syntax error

D) Also known as semantic error

Exam Questions:

- 2. By default, what a program does when it detects an exception?

A) Continue running

B) Results in the termination of the program

C) Calls other functions of the program

D) Removes the exception and tells the programmer about an exception

Exam Questions:

- 3. Why do we need to handle exceptions?

A) To avoid unexpected behaviour of a program during run-time

B) To let compiler remove all exceptions by itself

C) To successfully compile the program

D) To get correct output

Exam Questions:

- 4. How Exception handling is implemented in the C++ program?

A) Using Exception keyword

B) Using try-catch block

C) Using Exception block

D) Using Error handling schedules

Exam Questions:

- 5. Which part of the try-catch block is always fully executed?

A) try part

B) catch part

C) finally part

D) throw part

Exam Questions:

- 6. Which of the following is an exception in C++?

A) Divide by zero

B) Semicolon not written

C) Variable not declared

D) An expression is wrongly written

Exam Questions:

- 7. What is the difference between error and exception?

A) Both are the same

B) Errors can be handled at the run-time but the exceptions cannot

C) Exceptions can be handled at the run-time but the errors cannot

D) Both can be handled during run-time

Exam Questions:

- 8. What are the different types of exceptions?

A) 1

B) 2

C) 3

D) 4

Exam Questions:

- 9. Which keyword is used to throw an exception?

A) try

B) throw

C) throws

D) except

Answer the questions:

1) A
2) B
3) A
4) B
5) C
6) A
7) C
8) B
9) B