**DAILY ASSESSMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **26/06/2020** | **Name:** | **Dhavala** |
| **Course:** | **C++ Programming** | **USN:** | **4AL17EC027** |
| **Topic:** | * **Module 6: Templates, Exceptions, Files** | **Semester & Section:** | **6TH SEM & A Section** |
| **Github Repository:** | **Dhavala27** |  |  |

|  |
| --- |
| **SESSION DETAILS** |
|  |

|  |
| --- |
| **Report** Exceptions Problems that occur during program execution are called exceptions. In C++ exceptions are responses to anomalies that arise while the program is running, such as an attempt to divide by zero.Throwing Exceptions C++ exception handling is built upon three keywords: try, catch, and throw. throw is used to throw an exception when a problem shows up. For example:int motherAge = 29; int sonAge = 36; if (sonAge > motherAge) { throw "Wrong age values"; } The code looks at sonAge and motherAge, and throws an exception if sonAge is found to be the greater of the two.Catching Exceptions A try block identifies a block of code that will activate specific exceptions. It's followed by one or more catch blocks. The catch keyword represents a block of code that executes when a particular exception is thrown. Code that could generate an exception is surrounded with the try/catch block. You can specify what type of exception you want to catch by the exception declaration that appears in parentheses following the keyword catch. For example: try { int motherAge = 29; int sonAge = 36; if (sonAge > motherAge) { throw 99; } }  catch (int x) { cout<<"Wrong age values - Error "<<x; } //Outputs "Wrong age values - Error 99" Exception HandlingException handling is particularly useful when dealing with user input. For example, for a program that requests user input of two numbers, and then outputs their division, be sure that you handle division by zero, in case your user enters 0 as the second number.int main() { int num1; cout <<"Enter the first number:"; cin >> num1; int num2; cout <<"Enter the second number:"; cin >> num2; cout <<"Result:"<<num1 / num2; }Working with Files Another useful C++ feature is the ability to read and write to files. That requires the standard C++ library called fstream. Three new data types are defined in fstream: ofstream: Output file stream that creates and writes information to files. ifstream: Input file stream that reads information from files. fstream: General file stream, with both ofstream and ifstream capabilities that allow it to create, read, and write information to files.  To perform file processing in C++, header files <iostream> and <fstream> must be included in the C++ source file.#include <iostream> #include <fstream>Closing a File When you've finished working with a file, close it using the member function close().#include <iostream> #include <fstream> using namespace std; int main() { ofstream MyFile; MyFile.open("test.txt");  MyFile << "Some text! \n"; MyFile.close(); } |

