**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **26/06/2020** | **Name:** | **Mamatha.m** |
| **Course:** | **C++** | **USN:** | **4AL16EC035** |
| **Topic:** | **Challenge and certificate** | **Semester & Section:** | **6 b** |
| **Github Repository:** | **Mamatha-m** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**  **260ne.JPG**  **26 two.JPG** |
| **Report**  **CHALLENGES**  Drag and drop from the options below to declare a class "MyClass", with corresponding constructor and destructor. Constructor initializes class's private integer member named "mem".   class MyClass { public: MyClass() {    mem = 12; }  ~MyClass()   {    }  private:  int mem; };  Fill in the blanks to declare a class "test" with a "foo()" public member function. Declare a pointer "myPtr" to "test" and call "foo()" via the pointer.  class test{    : void foo() {    } };     myPtr = new test(); myPtr  foo();  Rearrange the code to define a function "foo", which throws an exception with a value of "-100" if its parameter is greater than 999. Then "foo" catches its exceptions and prints "error!" to the screen.  void foo(int arg)  {  try { if (arg> 999) throw -100; }  catch (int x)  { cout<< "error!" <<endl; }  }  Drag and drop from the options below to enter two integers and print their division to the screen. Use try and catch blocks to handle division by 0.   try{ int a; int b; cin >> a >> b; if (b == 0)  throw 0; cout << a / b << endl; }  catch (int err) {  cout << "error" << endl; }  What is the output of the following code?  int f=1, i=2; while(++i<5) { f\*=i; } cout<<f;  OUTPUT 12  Drag and drop from the options below to declare a file object and an associated file "myfile.txt", and write "I work with files" to the file if the file is open. Otherwise, print "Error" to the screen.  #include <iostream> #include <fstream> int main() { ofstream fileObj("myfile.txt");  if fileObj. is\_open()) {  fileObj << "I work with files";  fileObj.close(); } else {   cout << "Error" << endl;   } }  **MODULE 9**  **CERTIFICATE**  c++certificate.JPG |