LAB 1: Introduction

***Task1: namespace***

1. Please execute the program lab1­-1. Please try to identify the scope of variable defined in namespace Complex.

|  |
| --- |
| // lab1-1.h  namespace Complex{      typedef struct{          double real;          double image;      }Cplex;        const double pi = 3.1416;      void showComplex(const Cplex &m);  } |

|  |
| --- |
| // lab1-1.cpp  #include <iostream>  #include "lab1-1.h"  namespace Complex{      void showComplex(const Cplex &m)      {          std::cout << m.real;          if (m.image < 0)             std::cout << m.image << "i" << std::endl;          else             std::cout << "+" << m.image << "i" << std::endl;      }  } |

|  |
| --- |
| // lab1-1-main.cpp  #include <iostream>  #include "lab1-1.h"    int main()  {      Complex::Cplex n;      n.real = 1 \* pi;      n.image = -0.5;      Complex::showComplex(n);      return 0;  } |

🡺 Please modify the compiler error.

1. Please modify lab1-1-main.cpp as following and execute the program again.

|  |
| --- |
| // lab1-1-main.cpp  #include <iostream>  #include "lab1-1.h"    using namespace Complex;    int main()  {      Cplex n;      n.real = 1 \* pi;      n.image = -0.5;      showComplex(n);      return 0;  } |

🡺 Can you differentiate the above two programs?

***Task2: Debugging***

There are two kinds of errors you’ll run into when writing C++ programs: compilation errors and runtime errors.

Compilation errors are problems raised by the compiler, generally resulting from violations of the syntax rules or misuse of types. These are often caused by typos and the like.

Runtime errors are problems that you only spot when you run the program: you did specify a legal program, but it doesn’t do what you wanted it to. These are usually more tricky to catch, since the compiler won’t tell you about them.

1. Please execute the program lab1­-2-1. Did your program compile? If so, what does it print? If not, what error message do you get?

|  |
| --- |
| // lab1-2-1  #include <iostream>  #include <cstdlib>    using namespace std;  int main()  {      int 3x = 123;      cout <<  3x << endl;      return 0;  } |

1. Please execute the program lab1­-2-2. Did your program compile? If so, what does it print? If not, what error message do you get?

|  |
| --- |
| // lab1-2-2  #include <iostream>  #include <cstdlib>    using namespace std;  int main()  {      for(int i=0,i<10,i++){          cout << i << endl;      }      return 0;  } |

1. Please execute the program lab1­-2-3. Did your program compile? If so, what does it print? If not, what error message do you get?

|  |
| --- |
| // lab1-2-3  #include <iostream>  #include <cstdlib>    using namespace std;    int main()  {      for(int i=0;i<10;i++){          for(int j=0;j<10;j++){              cout << i << ". Ans=";              cout << i << "\*" << j << "=" << i\*j << endl;        }      return 0;;  } |

1. Please execute the program lab1­-2-4. Did your program compile? If so, what does it print? If not, what error message do you get?

|  |
| --- |
| // lab1-2-4  #include <iostream>  #include <cstdlib>    using namespace std;    int main()  {      int v1=12, v2=36;      cout << "before swap: v1 = " << v1 <<" ,v2 = " << v2 << endl;      v1 = v2;      v2 = v1;      cout << "after swap: v1 = " << v1 <<" ,v2 = " << v2 << endl;      return 0;  } |

***Task3: Main function***

*(Reference :* [*https://en.cppreference.com/w/cpp/language/main\_function*](https://en.cppreference.com/w/cpp/language/main_function)*)*

A program shall contain a global function named **main**, which is the designated start of the program in hosted environment. It shall have one of the following forms:

|  |  |
| --- | --- |
| intmain**() {** *body* **}** | (1) |
| intmain**(**int *argc***,** char\* *argv*[]**) {** *body* **}** | (2) |

|  |  |  |
| --- | --- | --- |
| ***argc*** | - | Non-negative value representing the number of arguments passed to the program from the environment in which the program is run. |
| ***argv*** | - | Pointer to the first element of an array of argc + 1 pointers, of which the last one is null and the previous ones, if any, point to [null-terminated multibyte strings](https://en.cppreference.com/w/cpp/string/multibyte) that represent the arguments passed to the program from the execution environment. If argv[0] is not a null pointer (or, equivalently, if argc > 0), it points to a string that represents the name used to invoke the program, or to an empty string. |
| ***body*** | - | The body of the main function. |

The names of *argc* and *argv* are arbitrary, as well as the representation of the types of the parameters: int main(int ac, char\*\* av) is equally valid.

A very common implementation-defined form of main() has a third argument (in addition to argc and argv), of type char\*\*, pointing at [an array of pointers to the *execution environment variables*](https://pubs.opengroup.org/onlinepubs/9699919799/functions/exec.html).