



COURSE INTRODUCTION

COURSE OBJECTIVES



- **Use** different types of advanced data structure
- **Create** program that can store data in external file rather than the memory
- **Design** program with sorting methods
- **Apply** C++ programming
- Introduction to basic **OOP** (object-oriented programming)

LEARNING PLANNING

1. Introduction to algorithm
2. Basic data types and statements
3. Condition
4. Loop
5. Array
6. Sub-programs (functions)
7. Structure

semester 1

8. *Recursive*
9. File IO
10. Pointers
11. Linked Lists
12. Stacks and Queues
13. Sorting algorithms
14. Tree
15. Basic OOP

semester 2



C

C++

STRUCTURE OF C++ PROGRAMMING



The image shows a screenshot of a C++ development environment. On the left, a code editor window titled "Demo1.cpp" displays the following C++ code:

```
1 #include<iostream>
2 using namespace std;
3
4 main() {
5     cout<<"Hello everyone!\n";
6     cout<<"Welcome to C++ programming!\n\n";
7 }
8
```

A vertical green bar highlights the first line of code. To the right of the code editor, a red box contains the text "Test.cpp". Below the code editor is a terminal window showing the output of the program:

```
C:\Users\bouch\Desktop\12-Test\review\S2\Demo1.exe
Hello everyone!
Welcome to C++ programming!
```

1st program in C++ programming

EXAMPLE IN C++ PROGRAMMING

```
1 #include<iostream>
2 using namespace std;
3 main(){
4     string name;
5
6     cout<<"Hello world!\n";
7     cout<<"Welcome to C++ language!\n";
8     cout<<"What is your name?: ";
9
10    cin>>name;
11    cout<<"\tHi, "<<name<<endl;
12 }
```



```
Hello world!
Welcome to C++ language!
What is your name?: Bob
                    Hi, Bob
```

Figure 1: A basic C++ program overview

Using **cin** and **cout** to get input and display