1.

```
8
9  #include <iostream>
10
11 int main() {
12     std::cout << "my favorite course after week 1 is [ECE 150].";
13     std::cout << std::endl;
14     return 0;
15 }
16
```

2.

```
make all
Building file: ../src/a00Q02.cpp
Invoking: Cygwin C++ Compiler
g++ -O0 -g3 -Wall -c -fmessage-length=0 -MMD -MP -MF"src/a00Q02.d" -MT"src/a00Q02.o" -o "src/a00Q02.o" "../src/a00Q02.cpp"
../src/a00Q02.cpp: In function 'int main()':
../src/a00Q02.cpp:12:7: error: 'cout' is not a member of 'std'
  std::cout << "Hello World!";
       ^~~~
../src/a00Q02.cpp:13:7: error: 'cout' is not a member of 'std'
  std::cout << std::endl;
       ^~~~
../src/a00Q02.cpp:13:20: error: 'endl' is not a member of 'std'
  std::cout << std::endl;
                    ^~~~
make: *** [src/subdir.mk:20: src/a00Q02.o] Error 1

16:16:19 Build Failed. 4 errors, 0 warnings. (took 486ms)
```

The complier cannot find ‘cout’ and ‘endl’ in ‘std’

3.

```cpp
#include <iostream>

int main();

int days_in_week(){
    std::cout << "7 days in a week.";
    std::cout << std::endl;
    return 0;
}

int main(){
    std::cout << "Hello World!";
    std::cout << std::endl;
    days_in_week();
    return 0;
}
```

4. these statements will be operated in the order of 2 1 4 3 5.

5. Function declaration is to tell the complier there is a function has this name.

Function definition is the actual body of the function.

6.

123

1.23

'a'

"Hello World!"

True

7.

```
#include <iostream>

int main() {
    std::cout << "\\\\'\"";
    std::cout << std::endl;
    return 0;

}
```

8.

std::endl can flush the output buffer and make the program more stable.

9.

```
#include <iostream>

int main();

int main(){
    std::cout << "A\t1\t\\\t\n"
              <<"Hgah\t3\t150\n"
              <<"\"X\"\tb\tD\n";
    std::cout << std::endl;

}
```

10.

No. Because keyword Identifiers are reserved with special meaning. a function prototype cannot be defined by keyword identifiers.

11.

N2354353g3

L43284969

12.

1. it will return an error of "division by zero"
2. it will print 10
3. it will print 10

13. & 14.

```
#include <iostream>

int main();

int main(){
    int m = 3;
    int x = 4;
    int b = 5;

    std::cout <<"The output of mx+b is:"
              << m*x+b
              << std::endl;

}
```

15.

“Std::cout << a*x*x*x + b*x*x + 3; “ works

16.

```cpp
#include <iostream>

void print_circle_info(float radius);


int main() {
    print_circle_info(5.3);
    return 0;
}

void print_circle_info(float radius){

    std::cout << "the diameter of the circle is "
            << 2*radius <<"\n";
    std::cout << "the circumference of the circle is "
                << 2*3.14*radius <<"\n";
    std::cout << "the area of the circle is "
                    << 3.14*radius*radius <<"\n";
    std::cout << std::endl;
    return;
}
```

17. & 18.

```cpp
#include <iostream>

int main();

int main(){

    for(int i=0; i <7; i++){
        for(int j=0; j<8; j++){
            if ((i+j)%2==0){
                std::cout << "o ";
            }else{
                std::cout << "+ ";
            }
        }
        std::cout << "\n";
    }

    return 0;
}
```

19. & 20

```
2
3  #include <iostream>
4
5  double func(double x);
6
7  int main() {
8      std:: cout << func(3.5);
9      std:: cout << std::endl;
10     return 0;
11 }
12
13 double func(double x){
14     double m = 2.34;
15     double b = 532.2;
16     return m * x + b;
17 }
18
```

21.

```
#include <iostream>

double func(double x, double m, double b);

int main() {

    return 0;
}

double func(double x, double m, double b){
    return m * x + b;
}
```

22. sorry I don't want to input them.

23. to prevent printing the connected word “EliberthGilthoniel,”

24. it will print a new line (has the same function of “\n”) and flush the output buffer

25.

```
#include <iostream>

int main() {
    std :: cout << (2+35+320+1+5+32+9)/7;
    std :: cout << std::endl;
    return 0;
}
```

It will print 57.

26.

```
#include <iostream>

int main() {
    std :: cout << (2+35+320+1+5+32.00+9)/7;
    std :: cout << std::endl;
    return 0;
}
```

It will print `57.7143` because after adding `32.00`, the type of data will be upcasted to double, so that the result will be double.