

601.220 Intermediate Programming

Function overloading

Function overloading

C++ compiler can distinguish functions with same name but different parameters

```
// overloading1.cpp
```

```
1  #include <iostream>
2
3  void output_type(int)    { std::cout << "int" << std::endl; }
4  void output_type(float) { std::cout << "float" << std::endl; }
5
6  int main() {
7      output_type(1);    // int argument
8      output_type(1.f); // float argument
9      return 0;
10 }
```

```
$ g++ -o overloading1 overloading1.cpp -std=c++11 -pedantic -Wall -Wextra
```

```
$ ./overloading1
```

```
int
float
```

Function overloading

But it **cannot** distinguish functions with **same name & parameters** but different return types

```
// overloading2.cpp
```

```
1  #include <iostream>
2
3  int  get_one() { return 1; }
4  float get_one() { return 1.0f; }
5
6  int main() {
7      int i = get_one();
8      float f = get_one();
9      std::cout << i << ' ' << f << std::endl;
10     return 0;
11 }
```

```
$ g++ -o overloading2 overloading2.cpp -std=c++11 -pedantic -Wall -Wextra
overloading2.cpp:4:7: error: ambiguating new declaration of float get_one()
    float get_one() { return 1.0f; }
      ~~~~~~
overloading2.cpp:3:7: note: old declaration int get_one()
    int  get_one() { return 1; }
      ~~~~~~
```

Quiz!

What output is printed by the following code?

```
#include <iostream>
```

```
char f(int c) {  
    if (c % 2 == 0) { return 'X'; }  
    else           { return 'Y'; }  
}
```

```
int f(char c) {  
    return (c - '0') * 11;  
}
```

```
int main() {  
    std::cout << f('7') << "," << f(7) << std::endl;  
    return 0;  
}
```

- A. 77,X
- B. 77,Y
- C. X,77
- D. Y,77
- E. The code does not compile

Quiz - answers

What output is printed by the following code?

// overloading3.cpp

```
1  #include <iostream>
2
3  char f(int c) {
4      if (c % 2 == 0) { return 'X'; }
5      else           { return 'Y'; }
6  }
7
8  int f(char c) {
9      return (c - '0') * 11;
10 }
11
12 int main() {
13     std::cout << f('7') << "," << f(7) << std::endl;
14     return 0;
15 }
```

At line 13:

Symbols (Scope)	Values
(f('7'))(main))	77
(f(7))(main))	'Y'