CS170#09.2

Class Templates. More Examples

Vadim Surov

1

```
class A {
public:
   template<typename T> T out(T t)  {
      return t;
   }
};

int main() {
   A a;
   return 0;
}
```

2

```
#include <iostream>

class A {
public:
   template<typename T> T out(T t)  {
      return t;
   }
};

int main() {
   A a;
   std::cout << a.out(1);
   return 0;
}
```

3

```
#include <iostream>

template<typename U>
class A {
public:
   template<typename T> U out(T t)  {
      return U(t);
   }
};

int main() {
   A<char> a;
   std::cout << a.out(65);
   return 0;
}
```

4

```
#include <iostream>
template<typename U>
class A {
public:
   template<typename T> U out(T t);
};
template<typename U>
   template<typename T> U A<U>::out(T t) {
      return U(t);
   }
int main() {
   A<char> a;
   std::cout << a.out(65);
   return 0;
}
```

5

```
#include <iostream>

template<typename U, typename T>
class A {
public:
   U out(T t)   {
     return U(t);
   }
};

int main() {
   A<char, int> a;
   std::cout << a.out(65);
   return 0;
}
```

6

```
#include <iostream>

template<int N, typename U, typename T>
class A {
public:
   U out(T t)   {
      return U(t+N);
   }
};

int main() {
   A<1, char, int> a;
   std::cout << a.out(65);
   return 0;
}
```

7

```
#include <iostream>

template<int N=1, typename U, typename T>
class A {
public:
  U out(T t)   {
     return U(t+N);
  }
};

int main() {
  A<1, char, int> a;
  std::cout << a.out(65);
  return 0;
}
```

8

```
#include <iostream>
template<int N=1, typename U=char,
                      typename T=int>
class A {
public:
   U out(T t)  {
      return U(t+N);
   }
};

int main() {
   A<2, char, int> a;
   std::cout << a.out(65);
   return 0;
}
```

9

```
#include <iostream>
template<int N=1, typename U=char,
                       typename T=int>
class A {
public:
   U out(T t)   {
      return U(t+N);
   }
};

int main() {
   A<> a;
   std::cout << a.out(65);
   return 0;
}
```

10

```
#include <iostream>
template<int N=1, typename U=char,
                       typename T=int>
class A {
public:
  U out(T t)   {
     return U(t+N);
  }
};

int main() {
  A a;
  std::cout << a.out(65);
  return 0;
}
```

11

```
#include <iostream>
class A {
public:
   class B {
   public:
      char out(int t) {
         return char(t);
      }
   };
   B b;
};
int main() {
   A a;
   std::cout << a.b.out(65);
   return 0;
}
```

12

```
#include <iostream>
class A {
public:
   template<typename U, typename T> class B {
   public:
      char out(int t) {
         return char(t);
      }
   };
   B<char,int> b;
};
int main() {
   A a;
   std::cout << a.b.out(65);
   return 0;
}
```