```
1 #include<iostream>
2
3 int main()
4 {
5     std::cout << "Hello Studygroup\n";
6 }
```

What will happen if we try to compile, link and run this program? Do you have any comments to the code?

```
1 #include <iostream>
2
3 int main() {
4     for (int i=0; i<3; i++)
5         std::cout << i;
6     for (int j=1; j<=3; ++j)
7         std::cout << j;
8     return 0;
9 }
```

What will this code print out?

foo.cpp

```
1  int main() {
2      int r = 42;
3
4      for (int i=1; i<8; ++i)
5          r += i;
6
7      return r;
8  }
```

bar.cpp

```
 1  int main() {
 2      int r = 42;
 3
 4      int i=1;
 5      while (i<8) {
 6          r += i;
 7          ++i;
 8      }
 9
10      return r;
11  }
```

Both these programs will return 70. But do you expect the resulting binary executable to be exactly the same? Eg,

```
g++ -S foo.cpp
g++ -S bar.cpp
diff foo.s bar.s
```

stack.h

```
1  namespace Stack {
2      void push(char);
3      char pop();
4
5      class Overflow {};
6      class Underflow {};
7  }
```

main.cpp

```
1  #include "stack.h"
2
3  int main() {
4      Stack::push('H');
5      Stack::push('e');
6      Stack::push('i');
7      Stack::pop();
8      Stack::pop();
9      Stack::push('e');
10     Stack::push('l');
11     Stack::push('l');
12     Stack::push('o');
13     return 0;
14 }
```

stack.cpp

```
1  #include <iostream>
2  #include "stack.h"
3
4  namespace Stack {
5      const int max_size = 4;
6      char v[max_size];
7      int top = 0;
8  }
9
10 void Stack::push(char c) {
11     if ( top == max_size )
12         throw Overflow();
13     std::cerr << "push: " << c << std::endl;
14     v[top++] = c;
15 }
16
17 char Stack::pop() {
18     if ( top == 0 )
19         throw Underflow();
20     char c = v[--top];
21     std::cerr << "pop:  " << c << std::endl;
22     return c;
23 }
```

```
g++ -Wall main.cpp stack.cpp && ./a.out
```

What will happen if you try to compile, link and run this code? Please comment this code.

stack.h

```
 1 class Stack {
 2     char * v;
 3     int top;
 4     int max_size;
 5
 6 public:
 7     class Underflow {};
 8     class Overflow {};
 9     class Bad_size {};
10
11     Stack(int s);
12     ~Stack();
13
14     void push(char c);
15     char pop();
16 };
```

main.cpp

```
 1 #include "stack.h"
 2
 3 int main() {
 4     Stack s(4);
 5     s.push('H');
 6     s.push('e');
 7     s.push('i');
 8     s.pop();
 9     s.pop();
10     s.push('e');
11     s.push('l');
12     s.push('l');
13     s.push('o');
14     return 0;
15 }
```

stack.cpp

```
 1 #include "stack.h"
 2 #include <iostream>
 3
 4 Stack::Stack(int s) {
 5     top = 0;
 6     if ( s < 0 || s > 16 )
 7         throw Bad_size();
 8     max_size = s;
 9     v = new char[s];
10 }
11
12 Stack::~Stack() {
13     delete v;
14 }
15
16 void Stack::push(char c) {
17     if ( top == max_size )
18         throw Overflow();
19     std::cerr << "push: " << c << std::endl;
20     v[top++] = c;
21 }
22
23 char Stack::pop() {
24     if ( top == 0 )
25         throw Underflow();
26     char c = v[--top];
27     std::cerr << "pop:  " << c << std::endl;
28     return c;
29 }
```

What will happen if you try to compile, link and run this code?
Please comment this code.

```
 1 #include <iostream>
 2
 3 class U {
 4     int a;
 5 };
 6
 7 class V {
 8     int a;
 9     int b;
10 };
11
12 class W {
13     int a;
14     int b;
15     void foo() { a = 3; }
16 };
17
18 class X {
19     int a;
20     int b;
21     void foo() { a = 3; }
22     void bar() { b = 4; }
23 };
24
25 class Y { };
26
27 class Z {
28     void foo() {}
29 };
30
31 int main() {
32     std::cout << sizeof(U) << std::endl;
33     std::cout << sizeof(V) << std::endl;
34     std::cout << sizeof(W) << std::endl;
35     std::cout << sizeof(X) << std::endl;
36     std::cout << sizeof(Y) << std::endl;
37     std::cout << sizeof(Z) << std::endl;
38 }
```

What will this code print out when executed?

```
 1 #include <iostream>
 2
 3 class U {
 4     int a;
 5     int b;
 6 };
 7
 8 class V : public U {
 9     int c;
10 };
11
12 class W : public V {
13 };
14
15 class X : public W {
16     int d;
17 };
18
19 class Y {};
20
21 class Z : public Y {
22     int e;
23 };
24
25 int main() {
26     std::cout << sizeof(U) << std::endl;
27     std::cout << sizeof(V) << std::endl;
28     std::cout << sizeof(W) << std::endl;
29     std::cout << sizeof(X) << std::endl;
30     std::cout << sizeof(Y) << std::endl;
31     std::cout << sizeof(Z) << std::endl;
32 }
```

What will this code print out when executed?

```
 1 #include <iostream>
 2
 3 class A {
 4 public:
 5     void foo() { std::cout << "A"; }
 6 };
 7
 8 class B : public A {
 9 public:
10     void foo() { std::cout << "B"; }
11 };
12
13 class C : public A {
14 public:
15     void foo() { std::cout << "C"; }
16 };
17
18 class D : public A {
19 };
20
21 void bar(A & a) {
22     a.foo();
23 }
24
25 int main() {
26     B b;
27     C c;
28     bar(b);
29     bar(c);
30 }
```

What will happen when you compile, link and execute this code?

```
 1 #include <iostream>
 2
 3 class U {
 4     int a;
 5     int b;
 6 };
 7
 8 class V {
 9     int a;
10     int b;
11     void foo() { a = 2; }
12     void bar() { b = 3; }
13 };
14
15 class W {
16     int a;
17     int b;
18     virtual void foo() { a = 2; }
19     virtual void bar() { b = 3; }
20 };
21
22 class X {
23     int a;
24     int b;
25     virtual void foo() { a = 2; }
26 };
27
28 int main() {
29     std::cout << sizeof(U) << std::endl;
30     std::cout << sizeof(V) << std::endl;
31     std::cout << sizeof(W) << std::endl;
32     std::cout << sizeof(X) << std::endl;
33 }
```

What will this code print out when executed?

```
 1 #include <iostream>
 2
 3 int main()
 4 {
 5     char ch = 'c';
 6     std::cout << '3';
 7     switch(ch) {
 8     case 'a':
 9         std::cout << 'a';
10     case 'b':
11         std::cout << 'a';
12     case 'c':
13         std::cout << 'c';
14     case 'd':
15         std::cout << 'd';
16     case 'e':
17         std::cout << 'e';
18     default:
19         std::cout << 'x';
20     }
21     std::cout << '4';
22     return 0;
23 }
```

What will this code print out?

```
 1 #include <iostream>
 2
 3 int main()
 4 {
 5     char ch = 'r';
 6     std::cout << '3';
 7     switch(ch) {
 8     case 'a':
 9         std::cout << 'a';
10     default:
11         std::cout << 'x';
12     case 'b':
13         std::cout << 'a';
14     case 'c':
15         std::cout << 'c';
16     }
17     std::cout << '4';
18     return 0;
19 }
```

What will happen when we try to compile, link and run this code?

```
 1 #include <iostream>
 2
 3 int main()
 4 {
 5     char ch = 'r';
 6     std::cout << '3';
 7     switch(ch) {
 8     case 'a':
 9         std::cout << 'a';
10         break;
11     case 'b':
12         std::cout << 'a';
13         break;
14     case 'c':
15         std::cout << 'c';
16         break;
17     defualt:
18         std::cout << 'x';
19         break;
20     }
21     std::cout << '4';
22     return 0;
23 }
```

What will this code print out when executed?

```
 1 #include <iostream>
 2 #include <list>
 3
 4 int main() {
 5     std::list<int> mylist;
 6     mylist.push_front(1);
 7     mylist.push_front(2);
 8     mylist.push_back(3);
 9     mylist.push_back(4);
10     for (std::list<int>::iterator i = mylist.begin();
11         i != mylist.end(); ++i) {
12         std::cout << *i;
13     }
14     return 0;
15 }
```

What will this code print out when executed?

```
 1 #include <iostream>
 2 #include <vector>
 3
 4 int main() {
 5     std::vector<char> v;
 6     v.push_back('a');
 7     v.push_back('b');
 8     v.push_back('c');
 9     v.push_back('d');
10
11     for (int i=0; i<v.size(); ++i) {
12         std::cout << v[i];
13     }
14     std::cout << std::endl;
15
16     for (std::vector<char>::size_type i=0; i<v.size(); ++i) {
17         std::cout << v.at(i);
18     }
19     std::cout << std::endl;
20
21     return 0;
22 }
```

Please comment this code.

```
 1 #include <iostream>
 2 #include <map>
 3 #include <string>
 4
 5 void print(std::pair<const std::string, int> & r) {
 6     std::cout << r.first << '/' << r.second << std::endl;
 7 }
 8
 9 int main() {
10     std::map<std::string, int> m;
11     m["A"] = 5;
12     m["E"] = 8;
13     m["B"] = 9;
14     m["C"] = 3;
15     for_each(m.begin(), m.end(), print);
16     return 0;
17 }
```

Please comment this code.

```
 1 #include <iostream>
 2 #include <map>
 3 #include <string>
 4 #include <algorithm>
 5
 6 bool gt_5(const std::pair<const std::string, int> & r) {
 7     return r.second > 5;
 8 }
 9
10 int main() {
11     std::map<std::string, int> m;
12     m["A"] = 5;
13     m["E"] = 8;
14     m["B"] = 9;
15     m["C"] = 3;
16     int c = count_if(m.begin(), m.end(), gt_5);
17     std::cout << c << std::endl;
18     return 0;
19 }
```

What will the following print out?

The third argument to `count_if` is called a predicate. Can you think of other ways of writing such predicates?