XC++02b - Functions, Parameters and Methods

Started: Sep 14 at 9:19am

Quiz Instructions

Question 1 — 1 pts

Functions in C and C++ pass all basic parameter types:

void f(int x) { ... }

- ○ By Reference
- ◉ By Value
- ○ By register
- ○ as ASCII text

Question 2 — 1 pts

```
void f(int x) {
  cout << x;

  x = 2;
}

int main() {
  int a = 5;
  f(a);
  f(a);
}
```

What is printed?

55

Question 3 — 1 pts

```
void f(int& x) {
  cout << x;
```

```
  x = 2;
}

int main() {
  int a = 5;
  f(a);
  f(a);
}
```

What is printed?

52

Question 4

1 pts

What is wrong with the following function call?

```
void f(int& x) {
  cout << x;
}

int main() {
  const int a = 5;
  f(a);
}
```

- a is an integer, but a pointer is being passed
- a is an integer, but a reference is being passed
- a is a constant, but a reference is passed
- there is nothing wrong with this code

Question 5

1 pts

What is wrong with the following function call?

```
void f(const int& x) {
  cout << x;
}

int main() {
  const int a = 5;
  f(a);
}
```

- a is an integer, but a pointer is being passed
- a is an integer, but a reference is being passed
- a is a constant, but a reference is passed
- (•) there is nothing wrong with this code

Question 6

1 pts

```
int main() {
  int a = 5;
  int&b = a;
  cout << b << " ";
  b = 4;
  cout << a;
}
```

What is printed?

5 4

Question 7

1 pts

```
int main() {

  int a = 5;
  const int& b = a;
  cout << a << b;
}
```

What, if anything, is wrong with this code?

- b is a constant but a is not
- (•) nothing is wrong
- Cannot have a reference in the same function as the value it references
- b is a pointer and not compatible with int

Question 8

1 pts

```
int main() {
  const int a = 5;
  int& b = a;
  cout << a << b;
}
```

What, if anything, is wrong with this code?

- (•) a is a constant but b is not
- ( ) nothing is wrong
- ( ) Cannot have a reference in the same function as the value it references
- ( ) b is a pointer and not compatible with int

Question 9

1 pts

```
int main() {
  const int a = 5;
  const int& b = a;
  cout << a << b;
}
```

What, if anything, is wrong with this code?

- ( ) a is a constant but b is a reference
- (•) nothing is wrong
- ( ) Cannot have a reference in the same function as the value it references
- ( ) b is a pointer and not compatible with int

Question 10

1 pts

```
int main() {
  const int a;
  float& b = a;
  cout << a << b;
}
```

What, if anything, is wrong with this code?

- [x] a is a constant but b is not
- [x] a is a const but not initialized
- [ ] the type of b does not match a

Question 11

1 pts

```
int main() {

  int a = 3;
  int& b = a;
  const int& c = b;
  b++;
  a--;
  cout << a << b << c;
}
```

333

Question 12

1 pts

What is output?

```
void f(int x) { cout << "hello" << x; }
void g(float x) { cout << "goodbye"; }
void g(double x) {}

int main() {
  f(1);
  g(1.0);
}
```

hellol

Question 13

1 pts

```
class A {
public:
```

```
  void f() { cout << "f"; }
  void g() const { cout << "g"; }

};

int main() {
  A a1;
  a1.f();
  a1.g();
}
```

fg

Question 14

1 pts

```
class A {
public:
  void f() { cout << "f"; }

  void g() const { cout << "g"; }
};

int main() {
  f();
}
```

- ○ The output is "f"
- ○ Illegal, the function is named A::f()
- ● Illegal, a method requires an object: a.f()

Question 15

1 pts

```
class A {
public:
  void f() { cout << "f"; }
  void g() const { cout << "g"; }
};


int main() {
  A a1;
  const A a2;
  a1.g();
  a2.f();
}
```

- ○ There is no problem
- ○ a1.g() is illegal, cannot call a const method on a non-constant object
- ◉ Illegal to call a2.f()

Question 16

1 pts

```
class A {

public:
  void g() { cout << "g"; }
  void f() const { g(); }
};

int main() {
  A a1;
  a1.g();
  a1.f();
}
```

What is wrong?

- ○ There is no problem
- ○ Method f is declared const, but changes the object
- ◉ Method f is declared const, but calls method g which is not

Question 17

1 pts

```
class Zebra {


private:
   int age;
   static int count;
public:
   Zebra() {
      age = 0;
      count++;
   }
```

```
    ~Zebra() {
        count--;
    }
    static int getCount() { return count; }
};

int Zebra::count = 0;

int main() {
        Zebra z1;
        Zebra z2;
        cout << Zebra::getCount() << "\n";
}
```

What is the output?

2

Question 18

1 pts

```
class Zebra {

private:
  int age;
  static int count;
public:
  Zebra() {
    age = 0;
    count++;
  }
  ~Zebra() {
    count--;
  }
  static int getCount() { return count; }
};
int Zebra::count = 0;
void f(Zebra z) {
  cout << Zebra::getCount();
}

int main() {
  Zebra z1;
  f(z1);
  Zebra z2;
```

```
  cout << Zebra::getCount();
}
```

What is the output?

11

Question 19 1 pts

```
class Zebra {

private:
  int age;
public:
  void print() const { cout << age; }
};

int main() {
  Zebra z1;
  z1.print();
}
```

What is the problem

- Every class must have a constructor
- The output is random
- z.print() is const

Quiz saved at 9:31am Submit Quiz