Midterm 2 Study Guide

Due No due date	**Points** 25	**Questions** 25	**Time Limit** 30 Minutes	**Allowed Attempts** Unlimited

Take the Quiz Again

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	30 minutes	18.33 out of 25

ⓘ Correct answers are hidden.

Submitted Jun 27 at 5:57pm

Question 1

1 / 1 pts

Which of these are ***guarded*** loops?

- [ ] `if`
- [ ] `if-else`
- [ ] `do-while`
- [x] `for`
- [x] `while`

Question 2

1 / 1 pts

Which are the two major categories of loops?

- [ ] infinite loops
- [ ] limit loops
- [ ] sentinel loops
- [x] definite loops
- [x] indefinite loops
- [ ] data loops

Question 3

1 / 1 pts

In the classic *for* loop, which portion of code is analogous to an *if* statement?

- None of these
- update expression
- first statement following the loop
- initialization statement
- assignment statement
- (•) condition expression

Question 4

1 / 1 pts

An ***unguarded*** loop is also known as a ***test-at-the-top*** loop.

- True
- (•) False

Question 5

1 / 1 pts

How many times is this loop ***entered***? (*That is, how many times is i printed?*)

```
for (int i = 1; i <= 10; i++)
    cout << i;
cout << endl;
```

- 9
- Never
- 11
- (•) 10

Question 6

1 / 1 pts

Match each item with the correct statement below.

Actions that occur after the loop is complete	postcondition
Actions occurring inside the loop's body	operation
Actions that occur before the loop is	

encountered	precondition
A test the determines if the loop should be entered	bounds

Question 7

1 / 1 pts

In the classic *for* loop, loop control variables going from 0 to less-than n are said to employ:

- ○ necessary bounds
- ○ intentional bounds
- ○ symmetric bound
- ◉ asymmetric bounds
- ○ None of these

Partial

Question 9

0.33 / 1 pts

Which of these are indefinite loops?

- ☐ counter controlled loops
- ☑ sentinel loops
- ☐ data loops
- ☐ limit loops

Question 10

1 / 1 pts

What is the output of the following?

```
bool token1 = true;
while (token1)
{
    for (int i = 0; i < 5; i++)
    {
        cout << "Hello there" << endl;
    }
    token1 = false;
}
```

- “Hello there” will be displayed infinite times.
- No output because of compilation error.
- No output.
- “Hello there” will be displayed 5 times.

Question 11

1 / 1 pts

Which prototypes in the following header file contain errors?

```
#ifndef EXAMPLE_H
#define EXAMPLE_H
#include <string>

string f1(int a);
int f2(double);
void f3(std::string& s, int n);
double f4();

#endif
```

- None of these
- *f4*
- *f3*
- *f2*
- *f1*

Question 12

1 / 1 pts

What prints here?

```
int i = 5;
while (i); cout << i--;
cout << endl;
```

- 4321
- Syntax error: `i` is not a Boolean expression
- 54321
- 43210
- (•) Infinite loop

Incorrect

Question 13

0 / 1 pts

Parameter names are optional in the function prototype.

- True
- (•) False

Incorrect

Question 14

0 / 1 pts

Assume that the input is `4 4 3 2 5`. What will print?

```
int i = 1;
int n;
cin >> n;
do
{
    i++;
    cin >> n;
}
while (n % 2);
cout << i << endl;
```

- 2
- 3
- Does not compile
- infinite loop
- (•) 4

Question 15

1 / 1 pts

What is the output of the following?

```
int i = 1;
int sum = 0;
while (i <= 13)
{
    sum = sum + i;
    i = i + 3;
}
cout << "The value of sum is " << sum;
```

- ○ The value of sum is 13
- ○ The value of sum is 22
- ◉ The value of sum is 35
- ○ The value of sum is 0

Question 16

1 / 1 pts

Different functions that have the same name, but take different arguments, are said to be:

- ○ covariant
- ○ default
- ◉ overloaded
- ○ derived
- ○ overridden

Incorrect

Question 17

0 / 1 pts

What prints here?

```
auto a = 3, b = 3;
cout << a == b ? "panda" : "tiger" << endl;
```

- ○ tiger
- ○ Does not compile
- ○ Undefined behavior
- ○ Crashes when run
- ◉ panda

Question 18

1 / 1 pts

Which line runs the `prt` program and stores its output in a new file named `x.data`?

- `./prt < x.data`
- **`./prt > x.data`**
- `./prt >1 x.data`
- `None of these`
- `./prt >> x.data`
- `./prt << x.data`

Question 19

1 / 1 pts

When running a filter program, you can send all output from `cout` to a file using the > redirection symbol.

- **True**
- False

Question 20

1 / 1 pts

Examine the code below:

```
int mystery1(int n, int a, int b) {
    if (n == 0) return a;
    if (n == 1) return b;
    return mystery1(n - 1, b, a + b);
}

int mystery2(int n) {
    return mystery1(n, 0, 1);
}
```

- [x] The algorithm implemented is Fibonacci
- [x] mystery1 is a recursive helper
- [ ] mystery1 is a recursive wrapper
- [x] if (n==1) is a base case

mystery2 is a recursive wrapper around the recursive helper mystery1. Together they implement the Fibonacci sequence in an efficient manner. mystery2 will not complete for any negative inputs.

Unanswered

Question 21

0 / 1 pts

Examine the code below and match the statements following it.

```
int mystery1(int n, int a, int b) {
    if (n == 0) return a;
    if (n == 1) return b;
    return mystery1(n - 1, b, a + b);
}

int mystery2(int n) {
    return mystery1(n, 0, 1);
}
```

mystery2 is a recursive wrapper	
mystery2 completes for all possible inputs	
if (n == 0) is a recursive case	

These functions illustrate how inefficient recursion is.

mystery2 is a recursive wrapper around the recursive helper mystery1. Together they implement the Fibonacci sequence in an efficient manner. mystery2 will not complete for any negative inputs.

Incorrect

Question 22

0 / 1 pts

Examine the code below:

```
int mystery1(int n, int a, int b) {
    if (n == 0) return a;
    if (n == 1) return b;
    return mystery1(n - 1, b, a + b);
}

int mystery2(int n) {
    return mystery1(n, 0, 1);
}
```

- [x] This is an inefficient way to implement this algorithm
- [ ] mystery2 completes for all inputs
- [x] mystery2 is a recursive wrapper
- [ ] if (n == 0) is a base case

mystery2 is a recursive wrapper around the recursive helper mystery1. Together they implement the Fibonacci sequence in an efficient manner. mystery2 will not complete for any negative inputs.

Question 23

1 / 1 pts

The C++ term for what is called a ***superclass*** in other languages is ***derived*** class.

- ( ) True
- (•) False

Question 24

1 / 1 pts

The file `grades.txt` contains lines of text that look like this:

```
Smith 94
Jones 75
. . .
```

Each line of text contains the student's name (a single word) and an integer score. What is the legal way of reading one student's information, given the following code?

```
string name;
int score;
ifstream in("grades.txt");
```

- None of these
- `in << name << score;`
- `getline(in, name); in >> score;`
- `in >> name >> score;`
- `getline(in, name); getline(in, score);`

Question 25

1 / 1 pts

Assume the user types "`brown  cow`" when this code runs. What prints?

```
char c;
cout.put(cin.get(c));
```

- `brown cow`
- `b`
- `true (or 1)`
- Does not compile