Midterm 2 Study Guide

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

Attempt History

	Attempt	Time	Score
KEPT	**Attempt 11**	29 minutes	25 out of 25
LATEST	**Attempt 13**	25 minutes	24 out of 25
	Attempt 12	30 minutes	22 out of 25
	Attempt 11	29 minutes	25 out of 25
	Attempt 10	30 minutes	20.17 out of 25
	Attempt 9	29 minutes	20 out of 25
	Attempt 8	29 minutes	20.5 out of 25
	Attempt 7	28 minutes	21 out of 25
	Attempt 6	24 minutes	21 out of 25
	Attempt 5	20 minutes	17.83 out of 25
	Attempt 4	30 minutes	17.67 out of 25
	Attempt 3	22 minutes	16 out of 25
	Attempt 2	16 minutes	16.17 out of 25
	Attempt 1	30 minutes	18.33 out of 25

▶
ⓘ Correct answers are hidden.

Submitted Jun 28 at 2:12pm

Question 1	1 / 1 pts

Loops are used to implement iteration in C++.

- ⦿ True
- ○ False

Question 2	1 / 1 pts

A **guarded** loop is also known as a **test-at-the-bottom** loop.

- ○ True
- ⦿ False

Question 3	1 / 1 pts

In the classic *for* loop, which portion of code is executed **after the last statement in the loop body**?

- ○ assignment statement
- ○ initialization statement
- ○ None of these
- ○ first statement following the loop
- ○ condition expression
- ⦿ update expression

Question 4
1 / 1 pts

Match each item with the correct statement below.

Keeps processing input until a particular value is found in input.
sentinel loop ▾

Keeps processing until the output gets no closer to the answer.
limit loop ▾

Repeats its actions a fixed number of times
definite loop ▾

Keeps processing until the input device signals that it is finished.
data loop ▾

Question 5
1 / 1 pts

Below is the illustration from the loop building strategy. The ***highlighted lines*** represent:

```
Given: the variable str is a string (may be empty)
Create the counter variable, initialized to -1
If the variable str has any characters then
{
    Set counter to 0
    Create the variable current-character as a character
    Place the first character in str into current-character
    While more-characters and current-character not a period
    {
        Add one to (or increment) the counter variable
        Store the next character from str in current-character
    }
    If current-character is a period then
        Add one to the counter to account for the period.
    Else
        Set counter to -2
}
If counter is -1 the string was empty
Else if counter is -2 there was no period
```

○ loop postcondition

○ loop bounds

○ bounds precondition

◉ goal operation

○ goal precondition

○ advancing the loop

Question 6
1 / 1 pts

In an ***unguarded*** loop, the loop actions may never be executed.

○ True

◉ False

Question 7
1 / 1 pts

The highlighted section below illustrates:

```
Given: the variable str is a string (may be empty)
Create the counter variable, initialized to -1
If the variable str has any characters then
{
    Set counter to 0
    Create the variable current-character as a character
    Place the first character in str into current-character
    While more-characters and current-character not a period
    {
        Add one to (or increment) the counter variable
        Store the next character from str in current-character
    }
    If current-character is a period then
        Add one to the counter to account for the period.
    Else
        Set counter to -2
}
If counter is -1 the string was empty
Else if counter is -2 there was no period
```

○ None of these

○ a postcondition

○ a loop guard

○ a boundary condition

○ a necessary condition

⦿ an intentional condition

Question 8   1 / 1 pts

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

```cpp
for (int i = 0; i < 10; i++)
    cout << i;
cout << endl;
```

⦿ 10

○ 9

○ Never

○ 11

Question 9   1 / 1 pts

Match each item with the correct question below.

What must I change in the test to go to the next iteration?	advance the loop
Can my loop reach its bounds?	necessary bounds
Has my loop reached its goal?	loop postcondition
What makes this loop quit?	loop bounds

Question 10   1 / 1 pts

Arguments passed to a function that has a constant reference parameter must be:

○ constants

◉ either lvalues or rvalues are fine

○ lvalues

○ reference arguments

○ rvalues

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

☑ *f1*

☐ *f2*

☐ *f4*

☐ *f3*

Question 12

1 / 1 pts

If a prototype in a header file has a parameter that is a library type, the header file must `#include` the appropriate library header.

◉ True

○ False

Question 13

1 / 1 pts

Header files may contain the statement `using namespace std;`

○ True

◉ False

Question 14

1 / 1 pts

To allow *f()* to change the argument passed here, the parameter *str* should be declared as:

```
void f( . . . str);
int main()
{
    string s = "hello";
```

```
        f(s);
}
```

○ const string

○ const string&

○ string

◉ string&

○ It is not possible for *f( )* to change the argument passed here.

Question 15

1 / 1 pts

What kind of error is this?

ex1.cpp:6:12: error: no viable conversion from 'int' to 'string'
**   string a = 15;**
**   ^   ~~**

○ Compiler error (something is missing when compiling)

○ Syntax error (mistake in grammar)

◉ Type error (wrong initialization or assignment)

○ None of these

○ Operating system signal or trap

○ Runtime error (throws exception when running)

○ Linker error (something is missing when linking)

Question 16

1 / 1 pts

Overloaded functions have the same name but different parameter names.

○ True

◉ False

Question 17

1 / 1 pts

Which prototypes in the following header file contain errors?

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

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

#endif
```

☐ *f4*

☐ *f1*

☐ *f3*

☑ None of these

☐ *f2*

Question 18

1 / 1 pts

When using the `get()` member function to read a character, leading whitespace is skipped.

○ True

⦿ False

Question 19

1 / 1 pts

What does this function do?

```
int mystery(int n, int m)
{
  if (n == 0) return m;
  return m * 10 + mystery(n / 10) + n % 10;
}
```

⦿ Computes the reverse of the input n

○ Computes the Factorial number n

○ Computes the Gauss series (sum) of 1..n

○ Computes the Fibonacci number n

○ Produces a stack overflow

Question 20

1 / 1 pts

What is the value of *r("xxhixx")*?

```
string r(const string& s)
{
  if (s.empty()) return "";
  if (s.at(0) == 'x') return 'y' + r(s.substr(1));
  return s.at(0) + r(s.substr(1));
}
```

○ Stack overflow

○ xyxyhixyxy

○ xxyyxx

○ yxyxhixyyx

⦿ yyhiyy

Question 21

1 / 1 pts

Which of the following is true about using recursion?

○ Recursion always helps you create a more efficient solution than other techniques.

○ None of the listed options.

◉ A recursive computation solves a problem by calling itself with simpler input.

○ A recursion eventually exhausts all available memory, causing the program to terminate.

Question 22   1 / 1 pts

What is the value of *r("axxbxx")*?

```
string r(const string& s)
{
    auto front = s.substr(0, 1);
    if (front.empty()) return "";
    return (front == "x" ? "" : front) + r(s.substr(1));
}
```

○ Stack overflow

◉ "ab"

○ "xxxx"

○ "a b "

○ "ax bx "

Question 23   0 / 1 pts

The file expenses.txt contains the line: `Hotel, 3 nights. $ 1,750.25`. What prints?

```
ifstream in("expenses.txt");
char c;
while (in.get(c))
{
    if (isdigit(c)) {
        in.unget();
        int n;
        in >> n;
        cout << n << 'x';
    }
}
```

○ 3x1x750x25

○ 3x (then cin fails)

◉ 3x1x750.25x

○ 3x1x750x25x

○ 3x1x7x5x0x2x5x

○ None of these

Question 24   1 / 1 pts

This command: `cat < nofile > /dev/null` will print an error message on the screen if `nofile` does not exist.

◉ True

○ False

Question 25

1 / 1 pts

Redirect standard error using the symbol 1> like this:

○ True

◉ False