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
KEPT	Attempt 11	29 minutes	25 out of 25
LATEST	Attempt 16	21 minutes	24 out of 25
	Attempt 15	30 minutes	23 out of 25
	Attempt 14	22 minutes	24 out of 25
	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 9:25pm

Incorrect **Question 1** 0 / 1 pts

Look at the problem statement below. The _____ of the loop is read a character and increment a counter.

> *How many characters are in a sentence? Count the characters in a string until a period is encountered. If the string contains any characters, then it will contain a period. Count the period as well.*

- bounds
- None of these
- plan
- goal

Question 2 1 / 1 pts

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

- True
- False

Question 3 1 / 1 pts

In a ***guarded*** loop, the loop actions may never be executed.

- (●) True
- ( ) False

Question 4

1 / 1 pts

Which line represents the ***necessary bounds*** in this loop?

```
1.      string s("Hello CS 150");
2.      while (s.size())
3.      {
4.         if (s.at(0) == 'C') break;
5.         s = s.substr(1);
6.      }
7.      cout << s << endl;
```

- ( ) None of these
- ( ) 5
- (●) 2
- ( ) 4

Question 5

1 / 1 pts

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

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

Question 6

1 / 1 pts

A loop that reads data until some special value is found is called a:

- ( ) data loop
- ( ) None of these
- ( ) limit loop
- ( ) loop and a half
- (●) sentinel loop

Question 7

1 / 1 pts

In the classic *for* loop, which portion is used to create the **loop control variable**?

- post-condition
- initialization statement
- update expression
- assignment statement
- condition expression
- None of these

Question 8

1 / 1 pts

Which are the two major categories of loops?

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

Question 9

1 / 1 pts

Which line ***advances the loop***?

```
1.      string s("Hello CS 150");
2.      while (s.size())
3.      {
4.         if (s.at(0) == 'C') break;
5.         s = s.substr(1);
6.      }
7.      cout << s << endl;
```

- 4
- 2
- 5
- None of these

Question 10

1 / 1 pts

Which of these are ***dependencies***?

```
EXE=digit-tester
OBJS=client.o digits.o
$(EXE): $(OBJS)
    $(CXX) $(CXXFLAGS) $(OBJS) -o $(EXE)
```

- [ ] $(EXE)
- [ ] None of these
- [ ] digit-tester
- [x] digits.o

- [x] `client.o`

Question 11

1 / 1 pts

Which of these prototypes is the best one to use in this circumstance?

```
int main()
{
    string str{"To be or not to be."};
    cout << "Most common letter is "
        << mostCommon(str) << endl;
}
```

- [ ] None of these are correct
- [ ] `char mostCommon(const string);`
- [ ] Any of these are fine.
- [ ] `char mostCommon(string&);`
- [ ] `char mostCommon(string);`
- [x] `char mostCommon(const string&);`

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.

- [x] True
- [ ] False

Question 13

1 / 1 pts

What kind of error is this?

```
~/workspace/ $ ./ex1
The Patriots won the 2018 Super Bowl
```

- [ ] Runtime error (throws exception when running)
- [ ] Syntax error (mistake in grammar)
- [ ] Operating system signal or trap
- [ ] Compiler error (something is missing when compiling)
- [x] None of these
- [ ] Type error (wrong initialization or assignment)
- [ ] Linker error (something is missing when linking)

Question 14

1 / 1 pts

Which of these are ***dependencies***?

```
EXE=digit-tester
```

```
OBJS=client.o digits.o
$(EXE): $(OBJS)
    $(CXX) $(CXXFLAGS) $(OBJS) -o $(EXE)
```

- [x] client.o
- [ ] $(EXE)
- [ ] digit-tester
- [ ] None of these
- [x] digits.o

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 35
- ( ) The value of sum is 0
- ( ) The value of sum is 13
- ( ) The value of sum is 22

Question 16

1 / 1 pts

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

```
void f( . . . str);
int main()
{
    f("hello");
}
```

- ( ) const string
- (•) const string&
- ( ) string
- ( ) It is not possible for *f()* to change the argument passed here.
- ( ) string&

Question 17

1 / 1 pts

What is the output of the following?

```
string s = "abcde";
int i = 1;
while (i < 5)
{
    cout << s.substr (i, 1);
```

```
    i++;
}
```

- No output
- abcde
- bcde
- abcd

Question 18

1 / 1 pts

Infinite recursion can occur because

- the recursive function is called more than once
- the recursive case is invoked with simpler arguments
- a second function is called from the recursive one
- the base case is missing one of the necessary termination conditions

Question 19

1 / 1 pts

What is the value of ***r(8818)***?

```
int r(int n)
{
    if (!n) return 0;
    return (n % 10 == 8) + (n % 100 == 88) + r(n / 10);
}
```

- 3
- Does not compile
- 4
- 1
- Stack overflow

Question 20

1 / 1 pts

How can you ensure that a recursive function terminates?

- Use more than one return statement.
- Provide a special case for the most complex inputs.
- Provide a special case for the simplest inputs.
- Call the recursive function with more complex inputs.

Question 21

1 / 1 pts

Which of the following symbol(s) can be used to redirect the output to a file or another program?

- [ ] <<
- [x] |
- [x] >>
- [x] >
- [ ] <

Question 22

1 / 1 pts

Unformatted I/O means that you read and write data line-by-line.

- ( ) True
- (•) False

Question 23

1 / 1 pts

One remarkably simple formula for calculating the value of π is the so-called Madhava–Leibniz series: $\frac{\pi}{4} = 1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\frac{1}{9}-\ldots$. Consider the recursive function below to calculate this formula:

```
double computePI(int number)
{
    if (number <= 1) { return 1.0;}
    int oddnum = 2 * number - 1;
    return  computesign(number) * 1.0 / oddnum
        + computePI(number - 1);
}
```

In this recursive function, what is the role of the helper function `computesign`?

- ( ) it checks the sign of the number and returns true if it is positive and false if negative
- (•) it makes sure the sign (positive or negative) alternates as each term of the series is computed
- ( ) it is called just one time to set the sign of the final result
- ( ) it is the recursive call in the function

Question 24

1 / 1 pts

What is the value of ***r(12777)***?

```
int r(int n)
{
    if (0 == n) return 0;
    int x = n % 10 == 7;    // 0 or 1
    return x + r(n / 10);
}
```

- ( ) Does not compile
- ( ) 2
- ( ) 5
- ( ) Stack overflow
- (•) 3

Question 25

1 / 1 pts

When using `cat` with redirection, the program stops when it runs out of input (a condition called end-of-file).

- (●) True
- ( ) False