**CISS240: Introduction to Programming**
**Quiz q0304**

Name: sithurman1@cougars.ccis.edu Score:

This is a closed-book, no compiler, 5 minute quiz.

Q1. In the following the repeating chunk of code repeats 4 times (i.e. it appears 5 times).

```
int i = 10, s = 0;

s = s + i;
i = i - 2;

s = s + i;
i = i - 2;

...
```

What is the final value of `i` and the final value of `s`? If you think the final value of `i` is `111` and the final value of `s` is `222`, write `111 222` with exactly one space between the two values.

Answer:

```
0 30
```

Q2. Which of the following are valid variable names? If (a),(b),(e),(f) are the only valid variable names, write `a b e f` in that order and with exactly one space between two consecutive letters.

(a) `wassup`
(b) `noway!`
(c) `as_far`
(d) `as3 far4`
(e) `5as_6far`
(f) `_as_far_`
(g) `gimme$`
(h) `RETURN`

Answer:

```
a c f h
```

Q3. T or F or M. For the following write T for true and F for false (ignore M).
The output of the following code fragment

```
int inheritance_from_uncle = 1000;
int amt_of_\$_in_wallet = 10;
int total_wealth = inheritance_from_uncle + amt_of_\$_in_wallet;
std::cout << total_wealth << '\n';
```

is

```
1010
```

Answer:

```
T
```

Q4. Here's a code fragment

```
int i = 0, j = 0;

j = j + i;
i = i + 1;

j = j + i;
i = i + 1;

j = j + i;
i = i + 1;
```

Note that there is a chunk of repeating code that appears 3 times. If the goal is to compute $0+1+2+3+4+5+6+7+8+9$ and store this value in variable `j`, how many times must the repeating chunk of code appear? If you think $0+1+2+3+4+5+6+7+8+9$ cannot be computed by repeat the above chunk of code, write ERROR.
Answer:

```
10
```

Instructions

In the file `thispreamble.tex` look for

```
\renewcommand\AUTHOR{}
```

and enter your email address:

```
\renewcommand\AUTHOR{jdoe5@cougars.ccis.edu}
```

(This is not really necessary since alex will change that for you when you execute `make`.) In your bash shell, execute "`make`" to recompile `main.pdf`. Execute "`make v`" to view `main.pdf`.

Enter your answers in `main.tex`. In the bash shell, execute "`make`" to recompile `main.pdf`. Execute "`make v`" to view `main.pdf`.

For each question, you'll see boxes for you to fill. For small boxes, if you see

```
1 + 1 = \answerbox{}.
```

you do this:

```
1 + 1 = \answerbox{2}.
```

`answerbox` will also appear in "true/false" and "multiple-choice" questions.

For longer answers that need typewriter font, if you see

```
Write a C++ statement that declares an integer variable name x.
\begin{answercode}
\end{answercode}
```

you do this:

```
Write a C++ statement that declares an integer variable name x.
\begin{answercode}
int x;
\end{answercode}
```

`answercode` will appear in questions asking for code, algorithm, and program output. In this case, indentation and spacing is significant. For program output, I do look at spaces and newlines.

For long answers (not in typewriter font) if you see

```
What is the color of the sky?
\begin{answerlong}
\end{answerlong}
```

you can write

```
What is the color of the sky?
\begin{answerlong}
The color of the sky is blue.
\end{answerlong}
```

A question that begins with "T or F or M" requires you to identify whether it is true or false, or meaningless. "Meaningless" means something's wrong with the question and it is not well-defined. Something like "$1 + 2 = 4$" is either true or false (of course it's false). Something like "$1+_2 = 4$?" does not make sense.

When writing results of computations, make sure it's simplified. For instance write 2 instead of $1 + 1$.

Higher level classes.

For students beyond 245: You can put LaTeX commands in `answerlong`.

More examples of meaningless statements: Questions such as "Is $42 = 1+_2$ true or false?" or "Is $42 = \{2\}^{\{3\}}$ true or false?" does not make sense. "Is $P(42) = \{42\}$ true or false?" is meaningless because $P(X)$ is only defined if $X$ is a set. For "Is $1 + 2 + 3$ true or false?", "$1 + 2 + 3$" is well–defined but as a "numerical expression", not as a "proposition", i.e., it cannot be true or false. Therefore "Is $1 + 2 + 3$ true or false?" is also not a well-defined question.

More examples of simplification: When you write down sets, if the answer is $\{1\}$, do not write $\{1, 1\}$. And when the values can be ordered, write the elements of the set in ascending order. When writing polynomials, begin with the highest degree term.

When writing a counterexample, always write the simplest.