

CISS240: Introduction to Programming
Quiz q0103

Name: bglandis1@cougars.ccis.eduScore:

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

Q1. For each part, identify the type of the value shown by writing S for string, C for character, and N for neither. If you think values are respectively character, string, string, neither, string, write your answer as C S S N S with exactly one space between two consecutive answers.

- (a) `"#include <iostream>"`
- (b) `'!'`
- (c) `'hello world'`
- (d) `"42"`
- (e) `"hello world`

ANSWER:

- (a) S
 - (b) C
 - (c) N
 - (d) S
 - (e) N

Q2. T or F or M.

When you run this program:

```
#include <iostream>

int main()
{
    std::cout << "the answer to life, universe, and everything is 42\n";

    return 0;
}
```

you will see the following output on your console window:

```
the answer to life, universe, and everything is 42\n
```

ANSWER:

F

Q3. What is the length of the following string "g\tr\te\te\tn"?

ANSWER:

9

Q4. T or F or M. The output of the following program

```
#include <iostream>

int main()
{
    std::cout << "_t/\t_\n";

    return 0;
}
```

is

t/\t

ANSWER:

F

Q5. Complete the program by printing ONE string so that it produces the given below:

```
#include <iostream>

int main()
{
    std::cout << _____;

    return 0;
}
```

Intended output:

1
2
3

ANSWER:

```
#include <iostream>

int main()
{
    std::cout << "1\n2\n3\n";
}
```

```
    return 0;  
}
```

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 L^AT_EX 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.