

## Quiz 2b

10 questions

---

1  
point

1.

In the following code, what does the number 100 represent?

```
my_button = frame.add_button("My Label", button_handler, 100)
```

Use the CodeSkulptor documentation to look it up.

- ☐ Vertical position of the button in pixels
  - ☐ Height of the button in pixels
  - ☐ Width of the button in pixels
  - ☐ Horizontal position of the button in pixels
- 

1  
point

2.

How many control objects are allowed in a frame?

- ☐ 0
- ☐ Unlimited, i.e., 0 or more
- ☐ 1

---

1  
point

3.

Which of the following are valid ways of defining and using a label?

Note that ellipses denote code that is omitted from and irrelevant for the question. Feel free to fill in plausible parameters and test each choice in CodeSkulptor.

☐

```
import simplegui

f = simplegui.create_frame(...)
label = f.add_label("My label")
label.set_text("My new label")
```

☐

```
import simplegui

frame = simplegui.create_frame(...)
frame.add_label("Label one")
frame.add_label("Label two")
```

☐

```
import simplegui

def label_handler():
    print "Handler called."

f = simplegui.create_frame(...)
f.add_label("My label", label_handler)
```

☐

```
import simplegui

simplegui.create_frame(...)
simplegui.add_label("My label")
```

---

1  
point

4.

When you enter text into an input field and press enter, the text is passed to the input field's event handler. What is the data type of the text?

- ☐ A string or a number, depending on the text entered
  - ☐ A string
  - ☐ A number
- 

1  
point

5.

Consider the following conditional statement.

```
if p == False:
    return False
elif q == False:
    return False
else:
    return True
```

That is equivalent to which of the following simpler statements?

Try to reason logically about each of the statements, but also try each in CodeSkulptor.

- ☐
  - ☐
  - ☐
  - ☐
-

point

6.

Which of the following describes the mistake in the following code?

```
def volume_cube(side):  
    """ Returns the volume of a cube, given the length of i  
    ts side. """  
    print side ** 3  
  
s = 5  
print "The volume of a cube with sides", s, "long is", volu  
me_cube(s), "."
```

- ☐ All of the printing should be done within the function.
  - ☐ The function should return, not print, its result.
  - ☐ The call to `volume_cube` shouldn't be within a `print` statement. More generally, function calls usually shouldn't be within `print` statements.
- 

1  
point

7.

What kind of errors can happen if you are missing a needed `global` declaration in one of your function definitions? For this question, you need only consider the case where the problem is in the function that is missing the `global` declaration.

If you are having trouble with this question, watch this week's Programming Tips video again.

- ☐ `AttributeError`
- ☐ An incorrect computation that generates no error message
- ☐ `NameError`
- ☐ `SyntaxError`

☐ Error: local variable '...' referenced before assignment

---

1  
point

8.

Which of the following function definitions are in the recommended code style?

☐

```
def f(x, y):  
    """ Add the two inputs. """  
    return x + y
```

☐

```
def f(x, y):  
    """ Add the two inputs. """  
    return x+y
```

☐

```
def myFunction(x, y):  
    """ Add the two inputs. """  
    return x + y
```

☐

```
def f (x, y):  
    """ Add the two inputs. """  
    return x + y
```

---

1  
point

9.

Cut and paste the following code into CodeSkulptor. Run it and make an attempt to understand how it works.

```
# Simple interactive application

import simplegui

# Define globals.

message = "Welcome!"
count = 0

# Define event handlers.

def button_handler():
    """Count number of button presses."""
    global count
    count += 1
    print message, " You have clicked", count, "times."

def input_handler(text):
    """Get text to be displayed."""
    global message
    message = text

# Create frame and register event handlers.

frame = simplegui.create_frame("Home", 100, 200)
frame.add_button("Click me", button_handler)
frame.add_input("New message:", input_handler, 100)

# Start frame.

frame.start()
```

We'd like to modify the code so that the count is reset to zero whenever a new message is entered. Where would you need to modify this code to implement this change?

- ☐ Add an assignment to `count` in the event handler for the input field. Also add a `global count` declaration there.
- ☐ Add an assignment to `count` in the initialization of global variables.
- ☐ Add an assignment to `count` in the event handler for the button.

☐ Add an assignment to count at the end of this code.

---

1  
point

10.

In the game “Guess the number”, what is the minimum number of guesses necessary to guarantee that the guesser can always win if the secret number is chosen in `range(0, 400)`?

Review the mini-project description for "Guess the number" if you are having trouble with this problem.

- ☐ 8 guesses
  - ☐ 9 guesses
  - ☐ 10 guesses
  - ☐ 12 guesses
  - ☐ It's impossible to guarantee that you can always win at “Guess the number”.
- 

6 questions unanswered

Upgrade to submit

