

**Question 1.** [6 MARKS]

Each of these subquestions contains a block of code. Treat each block of code independently (code in one question is not related to code in another), and fill in the blanks for each question.

**Part (a)** [1 MARK] **Order of Execution**

```
var_A = 11
var_B = var_A
var_A = 42
```

After this code is executed, the value of `var_B` is \_\_\_\_\_.

**Solution:**

11

**Part (b)** [3 MARKS] **Conditionals and Booleans**

The table to the right shows how an employee's age and experience affects his or her hourly wage. Assume that you have an `int` variable `experience` and an `int` variable `age` that correspond with the labels in the table. Fill in the boolean conditions in the code below to calculate the hourly wage for the employee.

Age	Experience		
	0	1-2	3+
under 18	\$6.50	\$9.50	\$11.00
18 and over	\$6.50	\$12.00	\$12.00

```

if _____:
    wage = 6.5

elif _____:

    if _____:
        wage = 9.5
    else:
        wage = 11
else:
    wage = 12

```

**Solution:**

```

if experience == 0:
    wage = 6.5

elif age < 18:

    if experience >= 1 and experience <= 2:
        wage = 9.5
    else:
        wage = 11
else:
    wage = 12

```

**Part (c)** [1 MARK] **Data Types**

Fill in the blank so that when this code is run, the user is asked to enter two numbers and then the sum of those numbers is printed. The user input may contain decimal values (e.g., 35000.75).

```
num1 = raw_input("Please enter your salary: ")
num2 = raw_input("Please enter your end of year bonus: ")

print "Your total earnings is" , _____
```

**Solution:**

```
float (num1) + float (num2)
```

**Part (d)** [1 MARK] **Calling Functions**

Fill in the blank to call `city_population` to obtain the population of Monkton in 1992.

```
def city_population(city, year):
    '''Return the population of str city in int year.'''
    # The code for this function is not shown.
    return population

print "In 1992, the population of Monkton was" , _____
```

**Solution:**

```
city_population ('Monkton', 1992)
```

**Part (e)** [0 MARK] **The Truth**

Who is the most captivating of this semester's three CSC108 instructors?

- A. Dan
- B. Dan
- C. Dan

**Question 2.** [6 MARKS]

Write the following function according to its docstring description.

```
def min_red (p):  
    '''p is a Picture object. Return the minimum red value  
    of all of the pixels of p. For example, if p has three pixels  
    whose red components are 100, 40, and 50, return 40.  
    p is guaranteed to have at least one pixel.'''
```

**Solution:**

```
import media  
  
def min_red (p):  
    '''p is a Picture object. Return the minimum red value  
    of all of the pixels of p. For example, if p has three pixels  
    whose red components are 100, 40, and 50,  
    return 40.'''  
  
    m = 255  
    for pixel in p:  
        m = min (m, media.get_red (pixel))  
    return m
```

**Question 3.** [8 MARKS]

In a certain dice game, a person's turn involves rolling two dice at a time, until:

- one of the dice shows a 6, or
- both dice show the same number

For example, here is a possible person's turn. (As above, there are two ways for a person's turn to end. Make sure you understand the other one, too.)

```
3 4
4 5
2 1
3 6
```

Write the following function so that it continues to roll two dice and print their values, until the person's turn is over. You'll want to generate a random number between 1 and 6 for each roll; you can use `random.randint (1, 6)` to do so. Note that each line of output should consist of two integers separated by a space, as in the sample given above.

```
def take_turn ():
    '''Print pairs of dice rolls until the turn is over.'''
```

**Solution:**

```
import random

def take_turn():
    '''Print pairs of dice rolls until the turn is over.'''

    prev_roll = random.randint (1, 6)
    cur_roll = random.randint (1, 6)
    while prev_roll != 6 and cur_roll != 6 and prev_roll != cur_roll:
        print prev_roll, cur_roll
        prev_roll = random.randint (1, 6)
        cur_roll = random.randint (1, 6)
    print prev_roll, cur_roll
```