Assignment 7: Code Reviews

Q1.

Select what feedback is **most** relevant to this piece of code

```
python
def bring(x, y)
  for e in x.keys():
    if e in y:
      x[e] = y[e]
```

A. This code is good!

B. This function does too many things and should be broken down

C. There is repetition that can be refactored

D. The names are poor and should be more descriptive

Q2.

Q2.1.

Select what feedback is **most** relevant to this piece of code

```
python
# Generate 3 random 3 digit pins
from random import randint
def generate_pins():
  result = []
  pin_1 = f'{randint(0,9)}{randint(0,9)}{randint(0,9)}'
  result.append(pin_1)

  pin_2 = f'{randint(0,9)}{randint(0,9)}{randint(0,9)}'
  result.append(pin_2)

  pin_3 = f'{randint(0,9)}{randint(0,9)}{randint(0,9)}'
  result.append(pin_3)

  return result
```

A. This code is good!

B. This function does too many things and should be broken down

C. There is repetition that can be refactored

D. The names are poor and should be more descriptive

Q2.2.

Your friend Njambi took a stab at rewriting this function. Consider her code carefully:

```
python
# Generate 3 random 3 digit pins
from random import randint
def generate_pins():
  return [f"{randint(0, 999)}" for i in range(3)]
```

What does this function do?

Q2.3.

In your opinion, is this a valid refactor of the original code? is the functionality the same? Justify your answer below

Q3.

Q3.1.

Select what feedback is **most** relevant to this piece of code

```
python
def notify_user_and_store_info(user_name, user_email, user_phone, user_street,
user_city, user_country):
   email = f"Dear {user_name},\nIt is a pleasure for us to welcome you to our
platform."
  email += f"We will connect you with other members in {user_city} over the next few
days."

  try:
    send_email(user_email, email)
  except:
    print(f"Failed to send email to {user_name} {user_email}")

  with open("user_storage.logs", "a") as record:
    user_address = f"{user_street}, {user_city}, {user_country}"
    record.write(f"{user_name}, {user_email}, {user_phone}, {user_address}
```

A. This code is good!

B. This function does too many things and should be broken down

C. There is repetition that can be refactored

D. The names are poor and should be more descriptive

Q3.2.

Describe what you could do to improve the code above. We are looking for at least two separate ways of making it better.

Share your ideas. You do not have to write any code, although you are welcome to.

Q4.

Consider the following code snippet from question 1:

```
python
def bring(x, y)
  for e in x.keys():
    if e in y:
      x[e] = y[e]
```

Your friend Adama tried to refactored it and came up with this version:

```
python
def update(original_dict, new_dict)
  for key, original_value in original_dict.items():
    original_dict[key] = new_dict.get(key, original_value)
```

Q4.1.

What does this function do?

Q4.2.

What feedback would you have for Adama?

A. This code is good!

B. This function does too many things and should be broken down

C. There is repetition that can be refactored

D. The names are poor and should be more descriptive