EECS 280 Midterm Exam
Spring 2023
Question Packet

This booklet contains questions and reference code. Answers go in a separate Answer Packet.

This is a closed-book exam. You may use one note sheet, 8.5"x11", double--sided, with your name on it.

Record all solutions in the Answer Packet. Nothing in this packet will be graded.

Turn in both this Packet and the Answer Packet. One will not be accepted without the other. Keep your note sheet. You may tear out the last piece of paper for scratch work.

Fill in multiple choice circles completely. Erase completely if you change your answer.

Assume all necessary `#include` headers and the `using namespace std;` directive are present unless otherwise indicated.

You do not need to verify `REQUIRES` clauses with `assert` unless indicated.

Assume all code is in standard C++11, and use only standard C++11 in your solutions.

Correctly completing the cover pages on this packet and the answer packet is worth **1 point**.

Signature: ______________________________

Name: ______________________________

Uniqname: ______________________________

Problem 1: Machine Model (20 Points)

1a) (5 points)

Refer to the following code snippet to answer the questions below.

```
1.    int x, y;
2.    int *z;
3.    int &w = x;
4.    x = 2;
5.    z = &y;
6.    y = x;
```

Answer the following statements with either true or false. [1 point each]

1. After line 1 executes, the values of `x` and `y` are undefined.
2. After line 2 executes, the value of `z` is `nullptr`.
3. After line 4 executes, what does the expression `x == w` evaluate to?
4. After line 5 executes, `z` holds the address of `y`.
5. After line 6 executes, `z` holds the address of `x`.

Refer to the following code snippet for questions **1b** and **1c**.

```
1.     int x = 4;
2.     int y = 3;
3.     int* const ptr = &x;
4.     // add line of code here
```

1b) Will the following line of code compile if it is placed at line 4? [1 point]

```
*ptr = 10;
```

1c) Will the following line of code compile if it is placed at line 4? [1 point]

```
ptr = &y;
```

Refer to the following code snippet for questions **1d** and **1e**.

```
int main() {
    int x = 1, y = 2;
    int &k = y;
    int *ptr = &y;
    k = x + y;
    cout << *ptr << endl;
    ptr = &x;
    *ptr = k + y;
    cout << x << endl;
    cout << k << endl;
}
```

1d) What is the output of this code snippet? [6 points]

1e) After this code snippet executes, what is the value of (`&y == &k`)? [1 point]

1f) (6 points)

What is the output of the following snippet of code?

```
int main() {
    int arr[] = {4, 2, 0, 3, 1};
    cout << *(arr + 2) << endl;

    int *ptr = arr + 1;
    ptr = ptr + *ptr;
    cout << *ptr << endl;

    int **ptr_ptr = &ptr;
    *ptr_ptr = ptr - **ptr_ptr;
    cout << *ptr_ptr - ptr << endl;
}
```

Problem 2: Procedural Abstraction and Testing (22 Points)

For questions **2a - 2c**, you can reference the abbreviated `Matrix` interface as needed below.

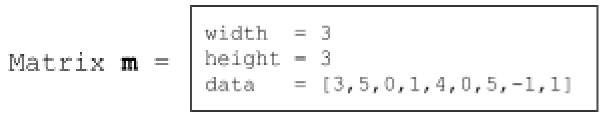
```
// REQUIRES: mat points to a valid Matrix
// EFFECTS:  Returns the width of the Matrix.
int Matrix_width(const Matrix* mat);

// REQUIRES: mat points to a valid Matrix
// EFFECTS:  Returns the height of the Matrix.
int Matrix_height(const Matrix* mat);

// REQUIRES: mat points to a valid Matrix
//           0 <= row && row < Matrix_height(mat)
//           0 <= column && column < Matrix_width(mat)
// EFFECTS:  Returns a pointer-to-const to the element in
//           the Matrix at the given row and column.
const int* Matrix_at(const Matrix* mat, int row, int column);
```

Read the following RME carefully before answering the following questions.

```
// REQUIES: mat points to a valid Matrix
//          0 <= col && col < Matrix_width(mat)
//          0 <= row_start && row_end <= Matrix_height(mat)
//          row_start < row_end
// EFFECTS: Returns the row of the element with the minimal value
//          in a particular region. The region is defined as elements
//          in the given column and between row_start (inclusive) and
//          row_end (exclusive). If multiple elements are minimal,
//          returns the row of the last occurrence (i.e. the
//          bottom-most one).
int Matrix_row_of_min_value_in_column(const Matrix* mat,
        int col, int row_start, int row_end);
```

2a) (3 points)

You are given a `Matrix` object `m` such that its member variables are initialized to the following:



```
Matrix m = width  = 3
           height = 3
           data   = [3,5,0,1,4,0,5,-1,1]
```

Given the RME above, what should the output of the following code snippet be?

```
    cout << Matrix_row_of_min_value_in_column(&m, 0, 0, 3) << endl;
    cout << Matrix_row_of_min_value_in_column(&m, 1, 1, 3) << endl;
    cout << Matrix_row_of_min_value_in_column(&m, 2, 0, 2) << endl;
```

Refer to the following implementation of the `Matrix_row_of_min_value_in_column` function, which compiles but contains **two** bugs, for questions **2b** and **2c**.

```
int Matrix_row_of_min_value_in_column(const Matrix* mat,
        int col, int row_start, int row_end) {
   1.    int min_row = 0;
   2.    int min = 0;
   3.
   4.    for (int row = row_start; row < row_end; row++) {
   5.        if (*Matrix_at(mat, row, col) < min) {
   6.            min_row = row;
   7.            min = *Matrix_at(mat, row, col);
   8.        }
   9.    }
   10.
   11.   return min_row;
}
```

2b) (5 points)

Given this implementation, what would be the output of the code snippet given in problem **2a**? The code snippet is shown again here for your convenience.

```
cout << Matrix_row_of_min_value_in_column(&m, 0, 0, 3) << endl;
cout << Matrix_row_of_min_value_in_column(&m, 1, 1, 3) << endl;
cout << Matrix_row_of_min_value_in_column(&m, 2, 0, 2) << endl;
```

2c) (6 points)

Using the table in the answer booklet, identify **two** lines that would have to change to fix the bugs and rewrite the lines to fix them.

Refer to the following function for the remaining parts of the question, which compiles but has a buggy implementation.

```
// REQUIRES: 'str' and 'substr' are valid, non-empty C-style strings
// EFFECTS:  Returns whether 'str' contains 'substr' (including the
//           null terminating character).
// EXAMPLES: has_substr("EECS280", "EECS") -> true
//           has_substr("-EECS--", "EECS") -> true
//           has_substr("EECS280", "RAND") -> false
bool has_substr(const char *str, const char *substr) {
   1.    while (*str) {
   2.        const char *a = str;
   3.        const char *b = substr;
   4.
   5.        while (*a == *b) {
   6.             a++;
   7.             b++;
   8.        }
   9.
   10.       if (!*b) {
   11.           return true;
   12.       }
   13.
   14.       ++str;
   15.   }
   16.
   17.   return false;
}
```

The bug is exposed when you call the function with the following arguments:

```
has_substr("EECS", "EECS");
```

2d) (3 points)

In the answer packet, briefly explain what happens in the `has_substr` function when it is run with the arguments above in no more than 2 sentences. Use line numbers for clarity in your answer.

2e) (5 points)

In the answer packet, use the table to identify the line number which reveals the bug and rewrite the line to fix it.

Problem 3: C-Style Programming and I/O (28 Points)

3a) (10 points)

You are creating a web application that requires users to create an account. You decide to use the following `struct` that represents a user account:

```
    const int MAX_LENGTH = 64;

    struct Account {
        char username[MAX_LENGTH];
        char password[MAX_LENGTH];
    };
```

To ensure the application is secure, implement the following C-style function, which *sanitizes* a username by removing a specified character.

```
// REQUIRES: 'acc' points to a valid Account object
// MODIFIES: 'acc'
// EFFECTS:  Removes all instances of char 'to_remove' from an
//           account's username
// EXAMPLE:  "!user!name!" -> "username" after removing "!"
void Account_sanitize_username(Account *acc, char to_remove);
```

3b) (6 points)

You are given a large number of user accounts and you want to make sure that all usernames are sanitized. Complete the function below, which takes a vector of account objects and uses the `Account_sanitize_username()` function to remove unsafe characters from each account's username.

```
// REQUIRES: none
// MODIFIES: 'accounts'
// EFFECTS:  Removes all characters in 'unsafe_chars' from each
//           username in the 'accounts' vector
// NOTE:     You must use Account_sanitize_username() in your solution
void sanitize_accounts(vector<Account> &accounts,
      const vector<char> &unsafe_chars);
```

3c) (12 points)

Implement a `main` function which reads from a file named `accounts.txt`, populates a vector with `Account` structs, and sanitizes all usernames in the vector.

The format of `accounts.txt` will be as follows:

```
    <Number of Unsafe Characters to Read>
    <Unsafe Character 1> <Unsafe Character 2> ... <Unsafe Character
N>
    <Username 1> <Password 1>
    <Username 2> <Password 2>
    ...
    <Username N> <Password N>
```

First, read in the unsafe characters into a vector, which will be used to sanitize the account usernames. Then, read each account into a vector until you reach the end of the file. Finally, use the `sanitize_accounts` function on the vectors you read in.

Here is a list of requirements and restrictions for this question:

- You **MUST** use the `sanitize_accounts` function in your solution.
- You **MUST NOT** use any standard library functions or declare any new variables in your solution.
- You may access the member variables of `Account` directly using the dot operator.
- You may assume that it is safe to read an unsafe character, username, and password using the extraction operator (`>>`). Recall that you can chain insertion operations together (ex. `cin >> x >> y`).

Problem 4: Polymorphism (22 Points)

4a) (6 points)

Refer to the answer packet for the class templates. Fill in the boxes in the code below so that:

- On their own, `Base` and `Derived` compile successfully.
- The code follows best practices for polymorphism and keywords such as `virtual` and `override`.
- The code in `main()` below behaves as described in the comments.
- If you believe a blank/box should be empty, write **BLANK** (do not just leave the box empty).

```
int main() {
  Derived d;
  d.print(); // prints "derived implementation" followed by a newline

  Base *b = &d;
  b->print(); // prints "base implementation" followed by a newline
}
```

For the remaining parts of problem 4, refer to the reference material below.

```
class Vehicle {
private:
    int tank_size;       // Current amount of fuel this Vehicle holds
    int tank_capacity;   // Max amount of fuel this Vehicle can hold

public:
    Vehicle(int tank_size_in, int tank_capacity_in)
    : tank_size(tank_size_in), tank_capacity(tank_capacity_in) {
        cout << "Vehicle ctor" << endl;
    }

    int get_tank_size() const {
        return tank_size;
    }

    virtual void describe() const {
        cout << "Tank Status: "
        << tank_size << "/" << tank_capacity << endl;
    }

    virtual void drive(int fuel) {
        tank_size = tank_size - fuel;
    }
};
```

```
class Motorcycle: public Vehicle {
public:
    Motorcycle(int tank_size_in)
    : Vehicle(tank_size_in, 50) {
        cout << "Motorcycle ctor" << endl;
    }
};

class Truck: public Vehicle {
public:
    Truck(int tank_size_in)
    : Vehicle(tank_size_in, 100) {
        cout << "Truck ctor" << endl;
    }

    virtual void drive(int fuel) = 0;
};

class FlatbedTruck: public Truck {
private:
    int cargo_weight;  // Weight of the cargo on the truck

public:
    FlatbedTruck(int tank_size_in)
    : Truck(tank_size_in), cargo_weight(0) {
        cout << "FlatbedTruck ctor" << endl;
    }

    void add_crate(int crate_weight) {
        cargo_weight = cargo_weight + crate_weight;
    }

    virtual void describe() {
        cout << "Cargo Weight: " << cargo_weight << endl;
    }

    // IMPLEMENT IN 4(c)
    virtual void drive(int fuel);
};
```

4b) (6 points)

Write a `main` function that produces the following output. If it is not possible to create the following output, write "NOT POSSIBLE" in the answer box.

```
Vehicle ctor
Truck ctor
FlatbedTruck ctor
Tank Status: 100/100
Cargo Weight: 50
```

4c) (4 points)

Implement the `FlatbedTruck::drive` function in the answer packet according to its RME.

```
// MODIFIES: 'tank_size' in the Vehicle class
// EFFECTS:  Subtracts the current 'tank_size' by both the 'fuel'
//           input AND by the flatbed truck's current 'cargo_weight'
virtual void FlatbedTruck::drive(int fuel);
```

4d) (3 points)

Consider the code snippet below:

```
int main() {
    FlatbedTruck ft(50);
    Truck *tr = &ft;
    // Add a line here
}
```

Which of the following lines could be added at the location indicated above with no compile errors? Select all that apply.

a. `ft.add_crate(10);`
b. `tr->add_crate(10);`
c. `cout << ft.get_tank_size() << endl;`
d. `cout << tr->get_tank_size() << endl;`
e. `ft.drive(5);`
f. `tr->drive(5);`

4e) (3 points)

Which of the following statements are true? Select all that apply.

a. If `m_ptr` is of type `Motorcycle *` and `t_ptr` is of type `Truck *`, the compiler will allow the assignment `m_ptr = t_ptr;`.
b. If `v_ptr` is of type `Vehicle *` and `t_ptr` is of type `Truck *`, the compiler will allow the assignment `v_ptr = t_ptr;`.
c. If `m_ptr` is of type `Motorcycle *` and `f_ptr` is of type `FlatbedTruck *`, the compiler will allow the assignment `m_ptr = f_ptr;`.
d. None of the above.

This Page Intentionally Left Blank

Do not write anything here, we will not grade it.