

Name:

Solution (S. Bassil)

CSIS 2610 Programming and Problem Solving S2025 Midterm Exam

Put your name on every page.

You have 75 minutes to complete this exam.

There are no books, notes, or resources of any kind allowed. **Put away all electronics, including computers, phones, etc.**

Anyone caught cheating will immediately receive a grade of zero for the entire course.

Section	Possible	Student
1. Function Design	15	
2. Loops	10	
3. Expression Evaluation	3	
4. Number Conversion	3	
5. Function Prototype and cin/cout Statements	9	
6. Tools	2	
7. Data Types	10	
8. More Function Design	8	
9. Tracing a Program	10	
Total	70	

Wednesday, February 26th, 2025

This page is meant to be blank.

SCENE 2: What's the best way to get to the airport?

There are no correct answers to this question.

Please write whatever you want.

Answer the following questions to determine the best way to get to the airport. You can choose one or more options.

Answer the following questions to determine the best way to get to the airport. You can choose one or more options.

1	Take the bus.	Bus number: 1
2	Take the train.	Train number: 2
3	Take the car.	Car number: 3
4	Take the plane.	Plane number: 4
5	Take the boat.	Boat number: 5
6	Take the bicycle.	Bicycle number: 6
7	Take the taxi.	Taxi number: 7
8	Take the motorbike.	Motorbike number: 8
9	Take the bicycle.	Bicycle number: 9
10	Take the bus.	Bus number: 10

Best way to get to the airport:

1. Function Design (15 points)

- (a) (6 points) Write a function (including body) called `isNotAlphanumeric` that accepts one character argument, `c`, and returns the value 1 if the argument is not a letter of the alphabet or a digit. The function returns the value -1 otherwise. Note that the decimal ASCII code for character 'A' is 65, the decimal ASCII code for character 'a' is 97, and the decimal ASCII code for character '0' is 48.

```

① int isNotAlphanumeric(char c)
{
    ① if ((a >= 48 && a <= 57) || (a >= 65 && a <= 90) ||
        (a >= 97 && a <= 122)) //c is alphanumeric
        ① return -1
    condition ①
}
① return 1

```

- (b) (5 points) Below is a slice of code that reads positive integer values from the existing non-empty file "`testdata`", until it runs out of values to read. The code calculates the average of these integers and saves the average to the output file "`result`". Here is an example: if values 13, 51, 7, 22 and 28 are read from `testdata`, the average 24.20 should be saved inside the output file "`result`". Write down all the things you see that are wrong with this code.

```

int x, sum = 0, count = 0, average;
ifstream f; // should be defined as float or double.
ofstream g; // (sum can be casted)
f.open("testdata");
g.open("result"); // missing.

if (f) {
    ③ f >> x; // this will read first value in file, but value not added to sum
    while (f >> x) {
        sum += x;
        count++;
    }
    average = sum / count;
    g << setprecision(2) << average;
}
else
    cout << "Error opening the input file.\n";
f.close();
g.close();

```

⑤ fixed manipulator is missing

(c) (4 points) Put the following items in order that they should appear in a C++ source code file:

- [5] function definitions
- [2] using namespace std;
- [1] include statements
- [3] function prototypes
- [4] main function

2. Loops (10 points)

For each of the following, determine how many times the function `event()` will be called, and the final value of the loop variable `i`.

*i = X2X4X8X8
event 11*

```
i = 1;
while (i++ < 5)
{
    event();
    i += 2;
}
```

`event()` was called how many times:

2

end value of `i`:

8

*i = XX2X2X4X56
event = 111*

```
for (i = 0; i <= 4; i++)
{
    event();
    i++;
}
```

`event()` was called how many times:

3

end value of `i`:

6

*i = XX2X32..
event 11...00*

```
for (i = 2; i++ < 3; i--)
    event();
```

`event()` was called how many times:

00

end value of `i`:

2 or 3

*i = XX2
j = XX3X23
Event 1111*

```
for (i = 0; i < 2; i++)
    for (j = 1; j < 3; j++)
        event();
```

`event()` was called how many times:

4

end value of `i`:

2

*i = XX4
event 11*

```
i = 2;
do
{
    event();
    i += 1;
} while (i < 4);
```

`event()` was called how many times:

2

end value of `i`:

4

3. Expression Evaluation (3 points)

Evaluate the following C++ expressions, giving their value.

$$1 \neq \frac{15 \% 7}{1}$$

False

$$(\neg 0 \& \& 0) \mid\mid (-1 \mid\mid !5)$$

T F
 F T
 F T

True

(for the following expression, assume $a = 2$)

$$\begin{array}{c} a++ + 1 \\ \hline 3 \end{array}$$

4. Number Conversion (3 points)

(a) Convert decimal value 101 into a one-byte *unsigned* binary.

$$\begin{aligned}
 101 &= 64 + 32 + 4 + 1 \\
 &= 2^6 + 2^5 + 2^2 + 2^0 \\
 &= 01100101 \leftarrow \text{answer}
 \end{aligned}$$

(b) Assume a computer with 8-bit integers. Convert the decimal integer -64 into a two's complement binary number.

$$\begin{array}{r}
 \text{abv}(-64) = 64 = 2^6 = 0100\ 0000 \\
 + 10111111 \\
 \hline
 1100\ 0000 \leftarrow \text{answer}
 \end{array}$$

(c) Assume a computer with 8-bit integers. Convert the two's complement binary integer 10110011 into a *signed* decimal.

$$\begin{array}{r}
 0100\ 1100 \\
 + 1 \\
 \hline
 0100\ 1101 = 2^6 + 2^3 + 2^2 + 2^0 = 64 + 8 + 4 + 1 \\
 = 77
 \end{array}$$

5. Function Prototype and `cin/cout` Statements (9 points)

- (a) For the following underlined function call, fill in the argument(s) type it expects, and provide the data type it is supposed to return. In other words, what does the function prototype look like?

(3 points)

```
const int CENTS = 100;
double number = 2.35;
double result;           double      int
result = doubleMe(number * 2 / CENTS);
```

argument(s) type:

double

returned data type:

double

function prototype:

double doubleMe(double);

- (b) (6 points) We are asking you to construct a `cout` statement that will print your name, in the order: your last name, a comma, a space, your first name, and leave the cursor at the beginning of the next line. E.g., the name *John Smith* would be printed as:

Smith, John

The black rectangle represents the final cursor position. You are given two *string* objects, `firstName` and `lastName`, which should contain your first name and last name, respectively. For that, prompt the user first to get values for those two objects, write one or more `cin` statements, then write the `cout` statement as described above. (Write your answer on the following page.)

① `(cout << "Input your first name: ";`
 ② `cin >> firstName;`
 ③ `(cout << "Input your last name: ";`
 ④ `cin >> lastName;`
`cout << lastName << [" "] << firstName << endl;`
 (1) (1) (1) (1)

6. Tools (2 points)

Answer the following two questions regarding compiling a C++ source code file.

(a) What does a **C++ compiler** do?

Convert source code into executable code. Detects errors.
 3 steps:
 • pre-processing
 • linking
 • ...

(b) What does the `g++ testFile.cpp -Wall` command do?
 Describe each flag.

-Wall enables warnings being specific about executable file's name

7. Data Types (10 points)

(a) (4 points) For each C++ data type below, fill in the table to answer:

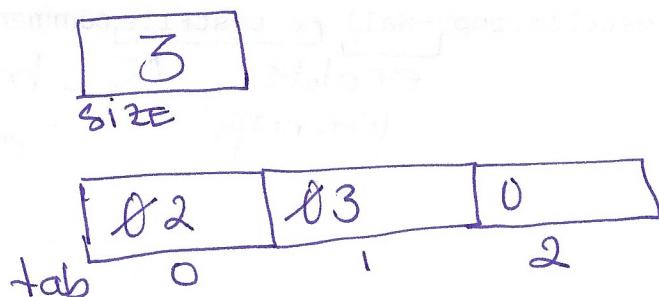
- How many bytes it requires (as discussed in class)?
- Is it signed? (Yes/No)
- What kind of encoding does it use (two's complement, ASCII chart, IEEE single or double precision standard)?

Data type	How many bytes?	Signed?	Encoding
short int	2	Y	2's compl
char	1	Y/N	ASCII chart
unsigned int	4	N	regular binary conversion.
float	4	Y	IEEE single Prece std.

(b) (6 points) Given the following piece of code in C++:

```
const int SIZE = 3;
int tab[SIZE] = {0, 0, 0};
for (int i = 0; i < 2; i++)
    tab[i] = i + 2;
```

Draw what the previous piece of code creates in memory. Use boxes to represent memory locations. Give full detail, including array name, subscripts, all the way down to data stored in each memory cell.



0x2

i

8. More Function Design (8 points)

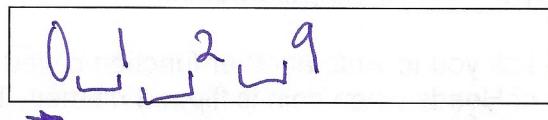
You are given a function called **coinFlipper**, which flips a coin (generates a random Heads or Tails). This function returns 0 if Heads is generated, it returns 1 if Tails is generated. **coinFlipper** expects no parameters.

In this exercise, we ask you to write another function called **headsCounter** that returns the number of Heads when coin is flipped n times. The value of n is provided as an argument to **headsCounter** and is guaranteed to be in the range 1 – 20, inclusive. **headsCounter** invokes the function **coinFlipper**.

```
int coinFlipper();  
  
int headsCounter(int n){  
    int count=0;  
    for(int i=0; i<n; i++){  
        if (coinFlipper() == 0)  
            count++;  
    }  
    return count;  
}
```

9. Tracing a Program (10 points)

Trace this program, show details on the next page, and record the output in the following box.



```
#include <iostream>
using namespace std;

int foo(int &n);

int main()
{
    const int VALUE = 6;
    int i, v;

    for (i = 0; i <= VALUE; i += 3) {

        v = i / 3;

        switch (v) {
            case 0:
                cout << "0 ";
                break;
            case 1:
                cout << "1 ";
                break;
            case 2:
                cout << "2 ";
        }
    }

    cout << foo(i) << "\n";

    return 0;
}

int foo(int &n)
{
    if (n < 8)
        return 0;

    return (n++);
}
```

In the following space, show what's happening at the memory level when the program on the previous page is executed. Use **boxes** to represent memory allocation, use **arrows** to represent function calls and control/data return from one function to the other, cross out boxes to represent memory de-allocation/release, etc. Give details as much as possible.

