McGill

April 2014
Final Examination

Introduction to Software Systems

COMP-206

April 15, 2013 at 18:00 – 21:00

Examiner: Joseph Vybihal Assoc Examiner:

Student Name:		McGill ID:									

INSTRUCTIONS:

- This is a **CLOSED BOOK** examination.
- You are permitted **TRANSLATION** dictionaries ONLY.
- **STANDARD CALCULATOR** permitted ONLY.
- This examination is **PRINTED ON BOTH SIDES** of the paper
- This examination paper **MUST BE RETURNED**
- You are permitted to write your answers in either **English or French**
- Write your answers in the **exam booklet** provided
- Attemp all questions, **part marks** will be assigned, show your work.

Grading

Section	Grade	Your Mark
Question 1: Short definitions	10	
Question 2: CGI	20	
Question 3: GNU	10	
Question 4 & 5: C	40	
Question 6: Python	20	
Total	100 %	

Points

QUESTION 1: SHORT DEFINITIONS

(1) 1. In this course we have been studying "Software systems". What is the primary acquired skill: to create an application distributed across multiple languages & run-time environments

(1) 2. Define the term "Client side": the programs running on the user's browser

(1) 3. Define the term "Back end": the programs running on the server

(1) 4. What is a "Packet": a data structure having from/to address & data. Used to communicate a msg over a network

(1) 5. What is a "Web server": a program that waits for a request from a client. Executes request. Returns answer to client.

(1) 6. What is a "Session": the period of time at the server between request arrival & the returned reply.

(1) 7. What is an "Interpreter": a program that executes a script without compilation

(1) Give one example: Bash, Python (not Java, C)

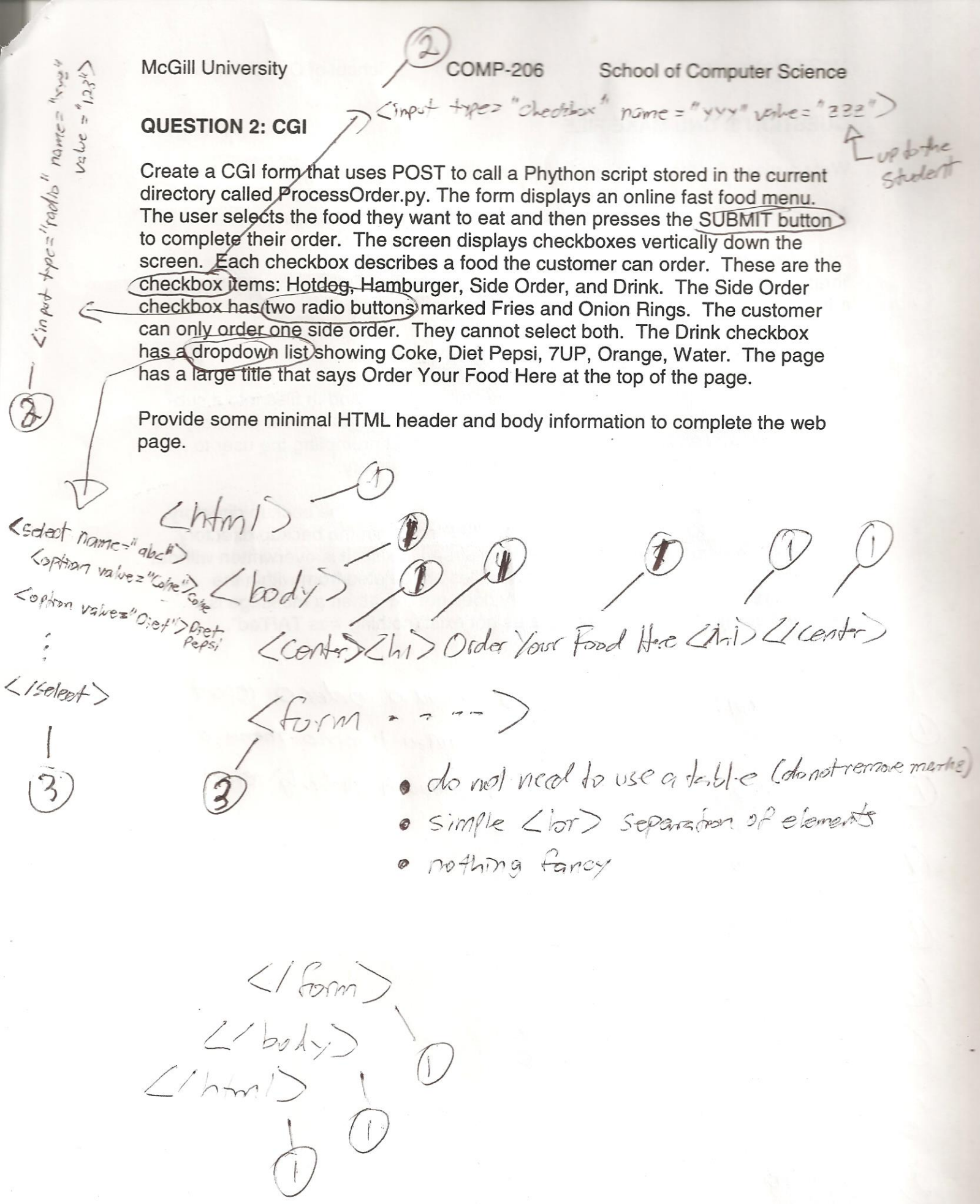
(1) 8. What is a "Compiler": a program that converts a source file into a binary file, directly executable by machine

(1) Give one example: C (maybe Java, not Bash, not Python)

QUESTION 2: CGI

Create a CGI form that uses POST to call a Phython script stored in the current directory called ProcessOrder.py. The form displays an online fast food menu. The user selects the food they want to eat and then presses the SUBMIT button to complete their order. The screen displays checkboxes vertically down the screen. Each checkbox describes a food the customer can order. These are the checkbox items: Hotdog, Hamburger, Side Order, and Drink. The Side Order checkbox has two radio buttons marked Fries and Onion Rings. The customer can only order one side order. They cannot select both. The Drink checkbox has a dropdown list showing Coke, Diet Pepsi, 7UP, Orange, Water. The page has a large title that says Order Your Food Here at the top of the page.

Provide some minimal HTML header and body information to complete the web page.



QUESTION 3: GNU MAKE FILE

Write a make file that handles this situation:

A programmer has the following project she wants to compile using make -

main.c and main.h where main.c includes library.h, menu.h, order.h, report.h
library.c and library.h
menu.c and menu.h
order.c and order.h where order.c includes menu.h, library.h
report.c and report.h where report.c includes library.h

The programmer also wants a target to backup all the .c and .h files into a sub-directory named backup. The directory is created within the current directory if it does not already exist. All the files are copied without prompting the user to confirm the copy of each file or the creation of the directory.

The programmer also wants a target to TAR all the files in the backup directory. All the files within the backup directory are TAR'ed within the backup directory using the name backup.tar. If that TAR file already exists it is overwritten without prompting the user. Then all the .c and .h files are deleted from within the backup directory. If the backup directory does not exist then a message is displayed saying "Backup directory does not exist, nothing was TAR'ed".

(1) program: main.o library.o menu.o order.o report.o
gcc -o program main.o library.o menu.o order.o report.o

(1) main.o : main.c library.h menu.h order.h report.h
gcc -c main.c

(1) library.o : library.c
gcc -c library.c

(1) menu.o : menu.c
gcc -c menu.c

(1) order.o : order.c menu.h library.h
gcc -c order.c

(1) report.o : report.c library.h
gcc -c report.c

(2) backup:
→ create dir if does not exist
→ copy *.c *.h w/o prompting
} any way student does this is okay in Bash

(2) tar:
→ if backup dir not exist then error msg & stop
else
→ tar all files into backup.tar w/o prompt
→ rm *.c *.h
} any way student does this is okay in Bash

QUESTION 4: C PROGRAMMING

Write a function having the following signature:
int parse (char line[], char buf1[], char buf2[], char buf3[], int bufSize)

The function parse() assumes the calling function sends a comma separated string within the array line[]. The size of the string can be derived from the end of string character. The calling function also provides three additional arrays of equal size. These arrays are empty. The size of these arrays are defined by the parameter bufSize.

→ -50% if a function is used.

To solve this problem you are not permitted to use any C library at all. You must do this with only C language commands: like arithmetic, loops, variables, arrays and/or pointers.

5 5

This function scans the string in array line[] extracting each field, without the comma, saving this into the arrays buf1[], buf2, and buf3. The function also counts the number of fields present in the input string line[]. This count is returned by parse().

1 3 1

Examples: 5
if line[] contains "A,B,C" then buf1 has "A", buf2 has "B:, buf3 has "C", returns 3
if line[] contains "A,B" then buf1 has "A", buf2 has "B", buf3 has "\0", returns 2
if line[] contains "A,B,C,D" then A, B, and C are in buf123, D is not, returns 4
Any number of fields can be in line[] separated by commas, not spaces.

eg/

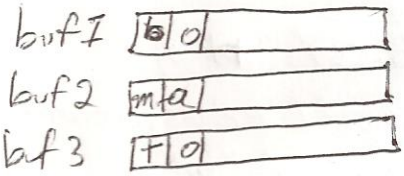
```
main()
{
   :
x=parse(line, buf1, buf2, buf3, 2);
   :
}
```

any size ends with \0

has this many spaces

if line has "bob, mary, tom, sam"

then



X = 4

QUESTION 5: C PROGRAMMING

Write a complete C program that begins by asking the user to enter the size of the array of students they would like to work with. The program then dynamically creates this array. The array is of type struct STUDENT and has the following fields: char name[50], int age, double gpa. The program then displays a text menu with the following options: (1) Add a student, (2) Delete a student, (3) Save all students, (4) Quit. The user stays in the menu until they select 4 to end the program. Option 1 asks the user to enter all the information for one student. It then asks for the cell number where the new student will be placed. If a student is already there it is overwritten without warning. The program does check if the cell number is valid. If it is not valid an error message is displayed and the user is prompted again to enter a cell number. This occurs until they enter a valid cell number. Option 2 only asks for a cell number. It validates that cell number and then deletes the information at the given cell number by assigning zero to the numbers and '\0' to the array. Option 3 writes all the occupied fields of the array as a comma-separated-vector file named students.csv. If the file already exists a warning message prompts the user to confirm the overwrite. If they say no then nothing is saved. If they say yes then the previous file is overwritten.

```
struct STUDENT *array = (struct STUDENT *) calloc( sizeof(...), n );
                                                          input by user

scanf("%s", name);
strcpy( *(array+i), name );
*(array+i).age = theAge;

do{
   scanf("%d", &i);
}while( i<0 || i>n );        if(i<0 || i>n) printf("error msg");

char ans = 'Y';
FILE *p = fopen("Students.csv", "wt");
if(p != NULL) { fclose(p); printf("exists, overwrite?"); scanf("%c", &ans); }
if(ans == 'Y'){ p = fopen("student.csv", "wt");
                for(i=0; i<n; i++)
                    fprintf(p, "%s, %d, %f \n", *(array+i).name, etc);
                fclose(p);
              }
```

QUESTION 6: PYTHON

(A) Provide a description only for those lines of code identified by a blank line on the right. (B) Then write what this program outputs and (C) what does it do.

(A) Describe this code

```
file = open("super_villains.txt")        — opens in read mode

name_list = []                           — empty list created
for line in file:                        — reads file line by line
    line = line.strip()                  — cleans ends of line
    name_list.append(line)               — adds to list

file.close()                             — close file

i = 0
while i < len(name_list) and name_list[i] != "Morgiana the Shrew":
    i += 1                               — count the cells, except ignore "Morgiana..."

if i == len(name_list):                  — Morgiana was not in the list
    print( "No." )
else:                                    — Morgiana was in the list, display yes & where she is (i)
    print( "Yes = ",i)

element = "Joker";                       — init Joker
low = 0
up = len(name_list)-1                    — init max size of list (counting from zero)
ff = False
while low <= up and ff == False:         — binary search looking for "Joker" in the list
    pos = (low + up) / 2                 — Calc centre position
    if name_list[pos] < element:         — Check centre lower than Joker
        low = pos + 1
    elif name_list[pos] > element:       — test higher than Joker
        up = pos - 1
    else:                                — if exists then ff = true else ff = false
        ff = True
if ff:                                   — if true,
    print(pos)                           — print Joker's position
```

(B) What does this program compute?

(1) Reads in a file of name
(2) Performs two searches
    a) linear search for Morgiana
    b) Binary search for Joker

(C) What does this program output? Assume the .txt files contains the names Joker, The Riddler, Cat Woman, Lex Luther, Magneto, Professor – formatted as needed by the program.

→ assume sorted alphabetically

Output:

No.
1