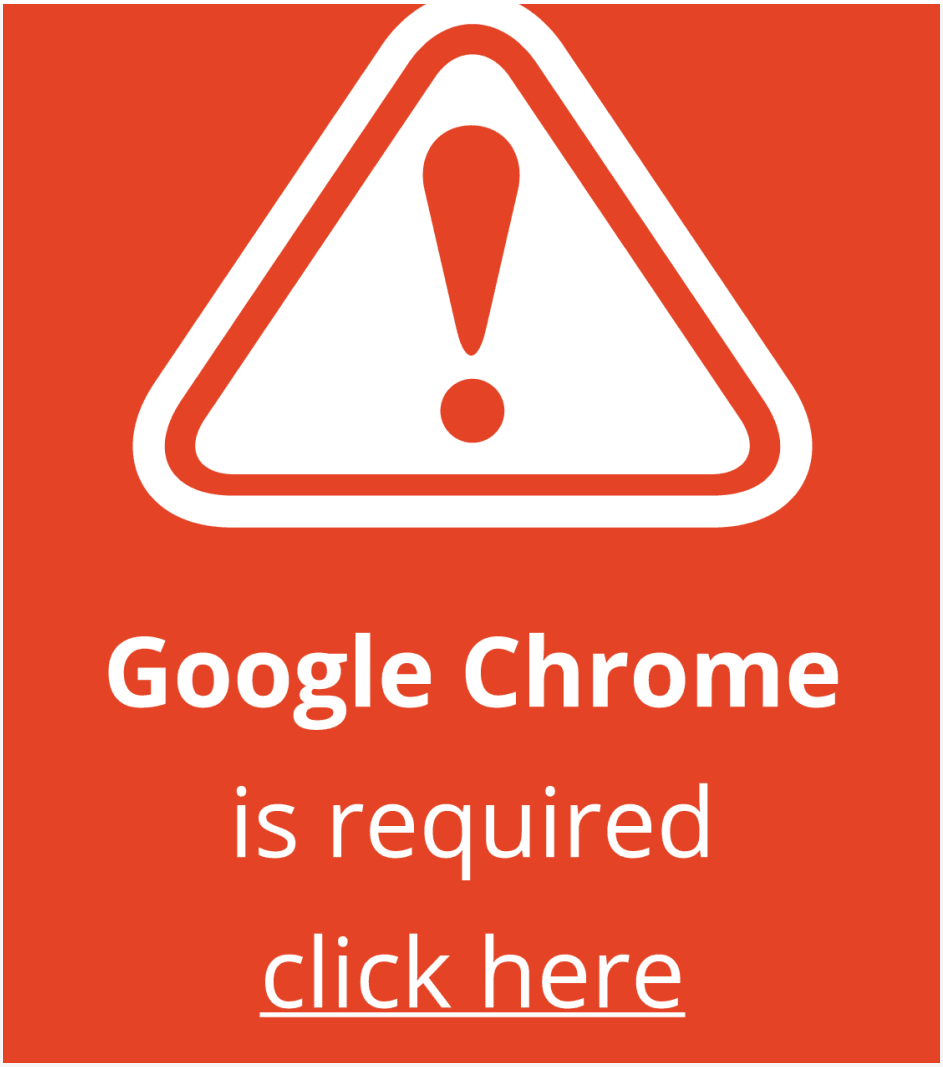
Review Test Submission: Exam-1 (Secure Browser)

User	Rod Miles Simms
Course	CS-1714-0B1-Fall-2020-Computer Programming II
Test	Exam-1 (Secure Browser)
Started	10/2/20 9:09 AM
Submitted	10/2/20 11:34 AM
Status	Completed
Attempt Score	109.5 out of 150 points
Time Elapsed	2 hours, 25 minutes out of 4 hours

Instructions



Results Displayed: All Answers, Submitted Answers, Correct Answers, Feedback, Incorrectly Answered Questions

Question 1 0 out of 4 points

What of the following is the correct usage of a **`typedef`** type for the following example?

```
typedef int * MyIntegerPointer;
```

Selected Answer:
```
int x = 5;
MyIntegerPointer * p = &x;
```

Answers:
```
MyIntegerPointer = 5;
```

```
int x = 5;
MyIntegerPointer * p = &x;
```

```
int x = 5;

typedef MyIntegerPointer p = &x;
```

```
   int x = 5;

MyIntegerPointer p = &x;
```

Question 2

4 out of 4 points

What is the output of the following code?

```
#include <stdio.h>
int fun(int *num)
{
    return (*num)--;
}
int main(int argc, char* argv[])
{
    int num = 16;
    for(fun(&num); fun(&num); fun(&num))
        printf("%d ", fun(&num));
    return 0;
}
```

Selected Answer: 14 11 8 5 2

Answers: 14 11 8 5 2

Infinite loop

13 10 7 4 1

15 12 9 6 3

Question 3

4 out of 4 points

If you have a **`struct`** to hold information about a person's name, phone number, and email address, which of the following declarations are correct?

Selected Answer:

```
typedef struct Person
{
    char name[ 100 ];
    int phone;
    char email[ 100 ];
} Person;
```

Answers:

```
typedef struct Person
{
    char name[ 100 ];
    int phone;
    char email[ 100 ];
} Person;
```

```
typedef struct Person
{
    int name;
    int phone;
    char email[ 100 ];
} Person;
```

```
typedef struct Person
{
    int phone;
    char email[ 100 ];
    char address[ 1000 ];
    char city[ 1000 ];
    int weight;
} Person;
```

```
typedef struct Person
{
```

```
    double amount;
} Person;
```

Question 4

4 out of 4 points

What functions can you use to have a user input a string?

Selected Answers:
- `scanf()`
- `fgets()`

Answers:
- `scanf()`
- `fgets()`
- `fscanf()`
- `input()`

Question 5

4 out of 4 points

In vim, in Command mode, what is the command to save the workspace and quit?

Selected Answer:
- ZZ

Answers:
- :qew
- zz
- ZZ
- :exit

Response Feedback: Since there was a typo in the options, you will receive full credit for this. The correct answer should be "ZZ".

Question 6

4 out of 4 points

For an array of 10 elements, named 'arr', which of the following is the correct way of accessing the 5th element?

correct way of accessing the 5th element?

Selected Answer: `arr[4]`

Answers: `arr[4]`

`arr(4)`

`arr(5)`

`arr[5]`

Question 7

4 out of 4 points

True or False: this code results in a syntax error because the loop goes beyond the last element of the array.

```
int i = 0;
int a[] = { 1, 5, 6, 10 };
for( i = 0; i < 100; i++ )
{
    printf( "%d\n", a[ i ] );
}
```

Selected Answer: False

Answers: False

True

Question 8

4 out of 4 points

What does the following code print out?

```
int x;
printf( "%d", x );
```

Selected Answer: Who knows? It's garbage.

Answers: 0

-1

Who knows? It's garbage.

Error: you can't print out a value without initializing it

Response Feedback: It should be 'garbage', but since I had mentioned this in class, I will give full credit if you mentioned '0'. But for the future (midterms, quizzes, final), it should be 'garbage' and no credit will be given to '0'.

Question 9

4 out of 4 points

What are the contents of the numbers array at the end of this code?

```
int numbers[] = { 35, 57, 78, 66, 41, 12 };

int *ptrA = numbers + 3;

int x = 15;

*ptrA = x++;
```

Selected Answer: **35, 57, 78, 15, 41, 12**

Answers:

25, 57, 78, 15, 41, 12

38, 57, 78, 66, 41, 12

35, 57, 78, 15, 41, 12

35, 57, 78, 66, 41, 12

Question 10

4 out of 4 points

What is the value of the **result** at the end of this code block?

```
char str[] = { '1', '2', '3', '4', '\0' };

int result = strlen( str );
```

Selected Answer: 4

Answers: 4

5

2

1

Question 11

18 out of 30 points

CS-1714-0B1-Fall-2020-Computer Programming II - aak759 - Rod Miles Simms

Write a function that prints the following pattern, based on the input value for N.

If N = 3, it should print

```
1 2 3
9 8 7
1 2 3
```

If N = 4, it should print

```
1 2 3 4
9 8 7 6
1 2 3 4
9 8 7 6
```

CS-1714-0B1-Fall-2020-Computer Programming II - aak759 - Rod Miles Simms

Selected Answer:

```
#include <stdio.h>
#include <stdlib.h>
void printSquareNums(int N){//function to print pattern
```

```
void printSquareNums(int N){//function to print pattern

   int i, j;//vars
   scanf("%d", &N);//input from user
   //loop through N rows
   for (i = 0; i < N; i++){
     for (i = 0; i < N; i++){
         for (j = 1; j <= N; j++){//print numbers 1-N
             printf("%d", j);
             printf("\n");//start newline
         }
         for (j = (i + 8); j >= (i +5); j--){//if N = 4 output 9, 8, 7,
6
            printf("%d", j);
         }
         printf("\n");
   }
}
```

Correct Answer:	[None]
Response Feedback:	should print different numbers in different rows as required for different N, not only for N=3 or N=4

Question 12

17 out of 30 points

CS-1714-0B1-Fall-2020-Computer Programming II - aak759 - Rod Miles Simms

Consider this block of code below. Trace the following code as explained in class. Show the starting address, the intermediate and final values of the variables. A variable of type 'int' takes 4 bytes and any pointer variable takes 8 bytes of space. Assume the starting address that is available is 1000 (calculate in decimal).

For each of the starting addresses S-1 to S-10, there is only one number, please write that down. However, for each of the variable values V-1 to V-10, there can be more than one values (starting, intermediate and final). Write all of them down, separated by comma. For example, if V-4 has 3 values then write them down as V-4 = 1, 2, 3.

You DON'T need to create a table for your answer. Just mention in each separate line, variable name = answer. For example,

S-1 = 1
S-2 = 2
and so on...

```
int X = 0, Y = 2, Z = 4;

int *p;

int **pp;

int arr[5] = { 1, 2, 3,
5, 8 };

p = &X;

pp = &p;

++*p;

p = &Y;

arr[++**pp++] = 30;

p = &arr[0];

arr[*p++] = 20;

*++p = 10;
```

Starting address location is 1000 (calculate in decimal)

Variable Name	Starting Address	Value
X	S-1	V-1
Y	S-2	V-2
Z	S-3	V-3
p	S-4	V-4
pp	S-5	V-5
arr[0]	S-6	V-6
arr[1]	S-7	V-7
arr[2]	S-8	V-8
arr[3]	S-9	V-9
arr[4]	S-10	V-10

Selected Answer:

S-1 1000 **V-1** 0
S-2 1004 **V-2** 2
S-3 1008 **V-3** 4

S-3 1008 **V-3** 4
S-4 1012 **V-4** 0, 1, 2
S-5 1020 **V-5** 0, 1, 2
S-6 1028 **V-6** 1, 20
S-7 1032 **V-7** 2, 30, 10
S-8 1036 **V-8** 3
S-9 1040 **V-9** 5
S-10 1044 **V-10** 8

Correct Answer: [None]

Response Feedback: v1, v2, v6, v7, v8, v9 partially correct, v4, v5 incorrect

Question 13

16.5 out of 20 points

CS-1714-0B1-Fall-2020-Computer Programming II - aak759 - Rod Miles Simms

Write code for the struct header file named "**course.h**" that consists of the following:

- **header guard**
- **struct definition** for **Course** which consists of 3 things:
  - **subject** (string of 2 characters) - refers to the subject/department for the course
  - **enrollment** (int) - refers to the number of students enrolled in the course
  - **GPA** (double) - refers to the average gpa obtained by all the students enrolled in the course
- **function prototypes** for the following 3 functions:
  - **fillCourses()** - The function will take in two parameters: array by reference and the length of the array. The function will not return anything. The function will initialize all the 3 data members for all the struct array objects.
    - **subject** values should be initialized to **"CS"**
    - **enrollment** values should be initialized to a **random value between 30 and 60, inclusive**
    - **GPA** values should be initialized to a the value **(index%9)/2.0**
  - **calculateAverageEnrollment()** - The function will take in two parameters: array by reference and the length of the array. It will calculate the average enrollment of all

courses, **without** considering the gpa of each course. This average value will be returned back by the function.

  - **calculateAverageOverallGPA()** - The function will take in

- **calculateAverageOverallGPA()** - The function will take in three parameters: array by reference, length of the array, and a double value (named **averageGPA**) passed as a pointer which will be updated with the calculated value. The function will not return anything. The **averageGPA** value should be updated to contain the weighted average of the gpa per course, by considering the enrollment of each course as well.

CS-1714-0B1-Fall-2020-Computer Programming II - aak759 - Rod Miles Simms

Selected Answer:

```
#ifndef COURSE_H
#define COURSE_H

typedef struct Course{
   char subject[2];
   int enrollment;
   double GPA;//avg of all students(enrollment +
[].enrollment0
}Course;
//prototypes
void fillCourses(int ref[]; int length);
int calculateAverageEnrollment(int ref [], int length);
void calculateAverageOverallGPA(int ref [], int length,
double *averageGPA);

#endif;//close header guard condition
```

Correct Answer: [None]

Response Feedback: subject length should be at least 3; the array in parameter should be a struct array
object; calculateAverageEnrollment() should return a double value

Question 14

17 out of 30 points

CS-1714-0B1-Fall-2020-Computer Programming II - aak759 - Rod Miles Simms

Write code for the struct source file named "**course.c**" that consists of the following:

- **include statements for all header files**
- **function definitions** for the following 3 functions:
    - **fillCourses()** - The function will take in two parameters: array by reference and the length of the array. The function will not return anything. The function will initialize all the 3 data members for all the struct array objects.
        - **subject** values should be initialized to **"CS"**
        - **enrollment** values should be initialized to a **random value between 30 and 60, inclusive**
        - **GPA** values should be initialized to a the value **(index%9)/2.0**
    - **calculateAverageEnrollment()** - The function will take in two parameters: array by reference and the length of the array. It will calculate the average enrollment of all courses, **without** considering the gpa of each course. This average value will be returned back by the function.
    - **calculateAverageOverallGPA()** - The function will take in three parameters: array by reference, length of the array, and a double value (named **averageGPA**) passed as a pointer which will be updated with the calculated value. The function will not return anything. The **averageGPA** value should be updated to contain the weighted average of the gpa per course, by considering the enrollment of each course as well.

CS-1714-0B1-Fall-2020-Computer Programming II - aak759 - Rod Miles Simms

Selected Answer:

```
#include <stdio.h>
#include <stdlib.h>
#include "course.h"
```

```
#include "course.h"

void fillCourses(int ref [], int length){
   int i, enrollment, GPA;
   for (i = 0; i < length; i++){
      strcopy(ref[i].subject, "CS");
      ref[i].enrollment = rand new random;(java)I don't know
how to psuedo random in c yet *_*
      ref[].GPA = (i % 9)/2.0;
   }
}
int calculateAverageEnrollment(int ref [], int length){
   int i;
   int avg = 0;
   int total = 0;
   for (i = 0; i < length; i++){
      total = total + ref[i].enrollment;
      avg = total / 2;
   }
   return avg;
}
void calculateAverageOverallGPA(int ref [], int length,
double *averageGPA){
  int i;
  *averageGPA = 0;//initialize before loop update in loop
   for (i = 0; i < length; i++){
      *averageGPA = ref[i].GPA/ref[i].enrollment;//avg
   }
}
```

Correct Answer:	[None]
Response Feedback:	string.h needed; the array in parameter should be a struct array, so type is Course; calculateAverageEnrollment() should return a double value; wrong enrollment range; wrong avg calculation; wrong averageGPA calculation;

Question 15

5 out of 10 points (Extra Credit)

CS-1714-0B1-Fall-2020-Computer Programming II - aak759 - Rod Miles Simms

Consider that the above Course files are already present and you are writing the code below in the main() function. Write code to do the following:

following:

- Create an integer 'N', which will be the number of courses.
- Take the value of 'N' from the user.
- Dynamically allocate memory for an array of 'N' Course objects.

CS-1714-0B1-Fall-2020-Computer Programming II - aak759 - Rod Miles Simms

Selected Answer:	#include <stdio.h>
	#include <stdlib.h>
	#include "course.c"
	
	int main (int argc, char *argv []){
	int N;
	scanf("%d", &N);//input from user
	(Course*)malloc(sizeof(Course)*N);//memory for *var from struct created * user number
	
	}

Correct Answer:	[None]
Response Feedback:	should allocate memory for an array

Monday, December 7, 2020 9:40:58 PM CST

← OK