# Parallel and MultiThreaded Programming

# CSYE 7215

# Homework 8

# Due: March 22, 2020

Put all your java, compiled class files and documentation into zip file named Homework8.zip and submit it via the dropbox on the blackboard before the END of due date. Put your name on all java files. There will be a short quiz on this assignment.

1. Write a program to create a int variable x with value 5.

a) declare int pointer variable ptr1

b) assign a valid address to ptr1

c) print x, &x, ptr1, \*ptr1

d) declare int pointer ptr2

e) assign ptr1 to ptr2

f) print x, &x, ptr1, \*ptr1, ptr2, \*ptr2

g) assign \*ptr1=9

h) print x, &x, ptr1, \*ptr1, ptr2, \*ptr2

i) compare (h) and (f), Explain

j) assign \*ptr2=15

k) print x, &x, ptr1, \*ptr1, ptr2, \*ptr2

l) i) compare (k) and (h), Explain

m) declare int array arr = {4, 12, 8, 6}

n) assign \*ptr1 to array arr

p) increment ptr1 four time, at each iteration, print ptr1, \*ptr, &arr, arr[i], ptr2, \*ptr2

q) Explain

2. Explain ElasticSearch concepts:

Cluster

Node

Index

Thread pools

Shards

Type

Document

ES uses threads to delete, update, insert (True/False)? How does it work?

How does ES handles when a Node in ES cluster goes down?

ElasticSearch use REST architecture (Yes/No)?What is Rest Architecture?

Spring MVC? How does it work?

Is ElasticSearch structured database or unstructured database? Give an example

What is document-oriented database? Give an example

What is Kabina?

ES uses threads to delete, update, insert (True/False)? How does it work?

How does ES handles when a Node in ES cluster goes down?

3. Use this article to install Java, Maven, Elasticsearch

<https://www.javacodegeeks.com/2018/03/elasticsearch-tutorial-beginners.html>

a) Implement application Steps: 1 through 7 described in the article.

b) Once (a) is working, add the following index “school” to Elasticsearch database.

c) Modify your program in (a) to ALSO read data “school” described below, and have

your program to report the data.

Create Index

PUT school

Response

{"acknowledged": true}

Add data

Request Body

POST school/\_doc/10

{

"name":"Saint Paul School", "description":"ICSE Afiliation",

"street":"Dawarka", "city":"Delhi", "state":"Delhi", "zip":"110075",

"location":[28.5733056, 77.0122136], "fees":5000,

"tags":["Good Faculty", "Great Sports"], "rating":"4.5"

}

Response

{

"\_index" : "school",

"\_type" : "\_doc",

"\_id" : "10",

"\_version" : 1,

"result" : "created",

"\_shards" : {

"total" : 2,

"successful" : 1,

"failed" : 0

},

"\_seq\_no" : 2,

"\_primary\_term" : 1

}

POST school/\_doc/16

{

"name":"Crescent School", "description":"State Board Affiliation",

"street":"Tonk Road",

"city":"Jaipur", "state":"RJ", "zip":"176114","location":[26.8535922,75.7923988],

"fees":2500, "tags":["Well equipped labs"], "rating":"4.5"

}

Response

{

"\_index" : "school",

"\_type" : "\_doc",

"\_id" : "16",

"\_version" : 1,

"result" : "created",

"\_shards" : {

"total" : 2,

"successful" : 1,

"failed" : 0

},

"\_seq\_no" : 9,

"\_primary\_term" : 7

}

4. Write declarations for the following:

a pointer to a character, an array of 10 integers, a reference to an array of 10 integers,

a pointer to an array of character strings, a pointer to a pointer to a character

a constant integer, a pointer to a constant integer, constant pointer to an integer

5. Write a function that swaps two integers. Use int\* as a argument type.

Write another swap function using int& as the argument type.

6. Consider the following data:

int nrows = 5;

int ncols = 10;

int row;

int \*\*rowptr;

Write a C++ program to create a two-dimensional array of rows and columns. Use the

following code to create columns.

for (row = 0; row < nrows; row++)

{

rowptr[row] = malloc(ncols \* sizeof(int));

if (rowptr[row] == NULL) {

printf("\nFailure to allocate for row[%d]\n",row);

exit(0);

}

printf("\n%d %p %d", row, rowptr[row], rowptr[row]);

if (row > 0)

printf(" %d",(int)(rowptr[row] - rowptr[row-1]));

}

7. Consider the following code. Print the values of xStart, xEnd, yStart, yEnd.

void offsetVector(double &x0, double &x1, double &y0, double &y1,

double offsetX, double offsetY) {

x0 += offsetX; x1 += offsetX; y0 += offsetY; y1 += offsetY;

}

void printVector(double x0, double x1, double y0, double y1) {

cout << "(" << x0 << "," << y0 << ") -> ("<< x1 << "," << y1 << ")" << endl;

}

main() {

double xStart = 1.2;

double xEnd = 2.0;

double yStart = 0.4;

double yEnd = 1.6;

offsetVector(xStart, xEnd, yStart, yEnd, 1.0, 1.5);

printVector(xStart, xEnd, yStart, yEnd);

}

8. Consider Linux Posix Thread program:

http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html

a) You can use “gcc” or “g++” compile to compile the program

>gcc example.cpp -o example OR >g++ example.cpp -o example

>example

b) Explain the code and Results

9. Consider the following defined structures. How do you print name, email,

title and year. Start with main().

struct movies\_t {

string title;

int year;

};

struct friends\_t {

string name;

string email;

movies\_t favorite\_movie;

} charlie, maria;

friends\_t \* pfriends = &charlie;

10. Consider the following code. Print all values of print\_all.

#include <iostream>

using namespace std;

void increment\_all (int\* start, int\* stop)

{

int \* current = start;

while (current != stop) {

++(\*current); // increment value pointed

++current; // increment pointer

}

}

void print\_all (const int\* start, const int\* stop)

{

const int \* current = start;

while (current != stop) {

cout << \*current << '\n';

++current; // increment pointer

}

}

int main ()

{

int numbers[] = {10,20,30};

increment\_all (numbers, numbers+3);

print\_all (numbers, numbers+3);

return 0;

}

11. Consider the following code. What is the value of “n”.

#include <iostream>

using namespace std;

int addition (int a, int b)

{ return (a+b); }

int subtraction (int a, int b)

{ return (a-b); }

int operation (int x, int y, int (\*functocall)(int,int)) {

int g;

g = (\*functocall)(x,y);

return (g);

}

int main () {

int m,n;

int (\*minus)(int,int) = subtraction;

m = operation (7, 5, addition);

n = operation (20, m, minus);

cout <<n;

return 0;

}