**NAME: EZRA KIPYEGON**

**UNIT: Systems Programming**

**UNIT CODE: CCS 3105**

**REG: C026-01-0739/2020**

**TASK: Assignment**

**Chapter 2**

***Chapter Summary Questions***

**1. In your own words, attempt to explain what a pointer really is, and what it is used for. If**

**you struggle to come up with a clean explanation that you wouldn’t want to show to**

**someone, try to reread some select sections of this chapter**.

A pointer is a variable that store the memory address location of another variable which has the same type as the pointer. For example a pointer of type integer store the memory address location of another variable of type integer.  
Implementation Example:

//code segment

int \*pointer , number = 12;

pointer = &number

//end of segment

**2. What is a memory allocation error? What are the different types of memory allocation**

**errors?**

A memory allocation error occurs when the active controller is low on memory after allocating resources and thus does not have enough remaining memory to control a stack member.

Some memory allocation errors include:

Memory leaks

This error occurs when there are objects in memory that are no longer in use by the program but are still being held.

Premature frees

This type of error is caused when an object that is still in use by the program is freed.

Double frees

This type of error occurs by trying to free an already freed object.

Wild frees

This error occurs when a program attempts to free a pointer in memory that was not returned by malloc ()

Memory smashing

This type of error occurs when less memory is allocated than the one that will be used.

**3. What is gdb? Why would you use it?**

gdb is a GNU debugger for projects based in the C and C++ language.

I would use gdb to find bugs in my program and use it to help in debugging.

**Chapter 3**

***Chapter Summary Questions***

**1. What command would you use to see the contents of a directory?**

The ‘ls’ command is used to list a directory’s content.

This command can take in flags such as -a to list also list hidden files.

**2. How do you provide the output of one command to another as input with a unixls**

**command?**

Using the pipe symbol, “|”. One can immediately call a second command after calling the first and provide the output of the first as input to the second command.

Example:

cat < test.txt | base64 > based64encodedfile.txt

//Explanation

The contents of the test.txt file is passed into the cat command.

The output of the cat command is then passed as the input of the base64 command

The output of the base64 command is then passed into the file named based64encodedfile.txt

**3. How do you access the arguments passed into a script?**

Inside the script, we can use the $ symbol followed by the integer to access the arguments passed.

For example:

Name of file : test.sh

echo "Your name is $1."

Execution:

chmod u+x ./test.sh && ./test.sh RandomName

Expected output:

Your name is RandomName

**4. Create an if else statement containing a while loop in bash, it doesn’t matter what they**

**do.**

**echo -n "Enter your age"**

**read AGE**

**if [ $AGE -gt 0 ]**

**then**

**for AG in {1..10};**

**do**

**echo "You were once $AG years old"**

**done**

**else**

**echo "The age input should be greater than zero"**

**fi**

**5. Write a regular expression to match both of these strings, do not use “.\*”: Aaab123 and**

**123Baaa.**

/A{0,1}a{0,2}b{0,1}123B{0,1}a{0,3}/gm

//Some explanation based on research

{n,y} is used to provide the limits on how many times a character should appear for example in this case n is the least number of times a character should appear and y is the maximum number the character should appear   
A{0,1} therefore means the minimum number A should appear can be zero and it can also appear one time

**Chapter 9**

***Chapter summary questions***

**1. Create a statement to insert James into our student database using the INSERT**

**statement. His ID is 6857938 and he is a sophomore. He is in the same college as you**

**are.**

INSERT INTO Students values (“James”, NULL, “6857938”, “SCIENCE”, “sophomore”);

**2. Use what you have learned about forming SQL statements to now delete James from the**

**database using the DELETE statement. This will work very similarly to the SELECT**

**statement.**

DELETE FROM Students WHERE StudentID = “6857938”;

**3. Now try using the UPDATE statement to change Dan’s Student ID to the number**

**1000000.**

UPDATE Students SET StudentID=”1000000” WHERE StudentID = “6857938”;

**4. Take the time to find an SQL connector for your favorite language online. A connector is**

**the library you will include that allows you to easily make connections to and send**

**statements to your SQL database from inside one of your programs**.

Language Used : Javascript

Package user : mysql as obtained in the npm package registry

//Connecting to the database

const connector = require("mysql").createConnection({

host: "localhost",

user: "root",

password: "",

database: "GIS",

});

connection.connect(error =>{

if(error) throw new Error(`Faced an error connecting to sql : ${error}`);

console.log("Connection to database made");

})

//

**Chapter 10**

***Chapter summary questions***

**1. Name a popular centralized source control system, and** **a popular distributed source**

**control system. Describe the difference between centralized and distributed source**

**control.**

**Difference**

In centralized version control, there is exactly one repository, and all contributors merely have access to it committing is an access to this possibly-remote repository. In distributed version control however, every contributor has one or more repositories of their own, and there need not be a separate repository in the centre committing is an entirely local operation.

An example of a popular distributed source control system is Git.

An example of a popular centralized source control system is Perforce , CVS , SVN

**2. Briefly describe the similarities between scrum and XP programming and describe which**

**one you personally would prefer and why.**

**Similarities**

They're both agile approaches.

They both dictate that work should be in an incremental way.

Both approaches emphasize the importance of team autonomy, transparent communication, and prioritizing work that delivers the most value

I would prefer scrum

Explanation on why I would prefer scrum over XP programming

I personally prefer Scrum because it's a framework that allows people to tackle complex adaptation problems while being productive and creative in delivering the highest possible value product.

**3. List the types of tests and when to use each.**

Unit tests

Unit Testing is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected. A unit may be an individual function, method, procedure, module, or object.

System Tests

System testing is the process in which a quality assurance team evaluates how the various components of an application interact together in the full, integrated system or application.

Regression Tests

Regression testing is testing existing software applications to make sure that a change or addition hasn't broken any existing functionality.

Acceptance Tests

Acceptance testing is done to determine whether or not the software system has met the requirement specifications. The main purpose of this test is to evaluate the system's compliance with the business requirements and verify if it is has met the required criteria for delivery to end users.

**4. Pick a few of the refactoring guidelines and look back at an old project you have worked**

**on and correct it to match the guidelines or other guidelines you come up with on your**

**own.**

Use of comments so as to vividly explain what a certain piece of code does.

Use of functions so as to avoid redundancy.

Use of short functions/code blocks so as to easily understand what is being done by the function.

Getting rid of code that does not work / dead code from the codebase.

Have consistent naming for variables and functions.