Assignment xx Algorithmic Design Document

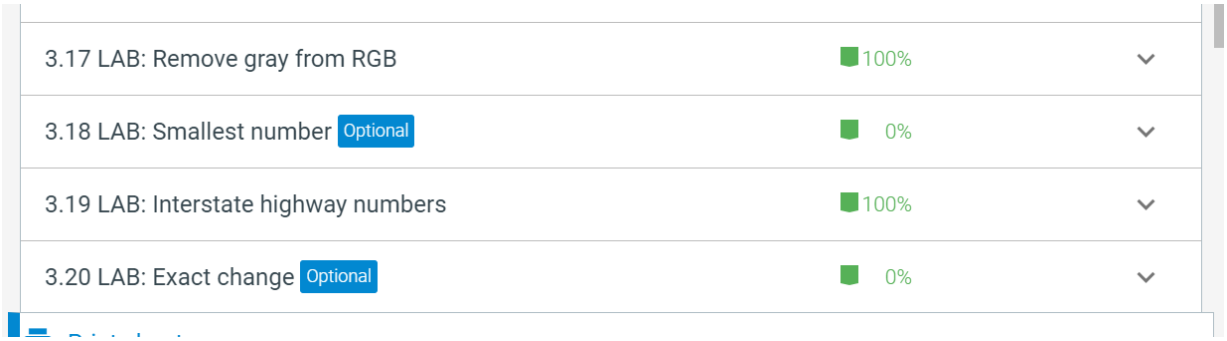
Make a copy before you begin (File -> Make a copy). Add the Assignment # above and complete the sections below BEFORE you begin to code and submit with your Assignment to D2L (File -> Download -> PDF). The sections will expand as you type.

zyBooks

Add your zyBooks screenshots for the % and assigned zyLabs completions below. Required percentages: all assigned zyLabs, Challenge Activity with at least 70%, and Participation Activity with at least 80%.

zyLabs, Challenge, and Participation % Screenshot:

Assigned zyLabs completion Screenshot:



Assignment

Program description:

This program will let you play the incredible game “Rock, Paper, Scissors!” with your computer. Just type in Rock, Paper, or Scissors and you’ll be competing with the computer immediately!

Before you begin coding, **you must first plan out the logic** and think about what data you will use to test your program for correctness. All programmers plan before coding - this saves a lot of time and frustration! Use the steps below to identify the inputs and outputs, calculations, and steps needed to solve the problem.

Algorithmic design:

a. Identify all of the user input. What are the data types of the inputs? Define the input variables.

userInput as STRING

b. Describe the program output. What is displayed to the user? What are the data types of the output? Define the output variables.

The program will output the following strings:

Lets play Rock, Paper, Scissors!

Please type "Rock", "Paper", or "Scissors" to choose your hand.
Note: You can type "R", "P", or "S" too.

Please choose your Hand:

You have chosen: {userInput}!
The computer has chosen: Paper

Paper beats {userInput}.
The computer has won!

Thank you for playing!

c. What calculations do you need to do to transform inputs into outputs? List all formulas needed, if applicable. If there are no calculations needed, state there are no calculations for this algorithm.

Arithmetic:
srand((int)time(0)) //Random seed declaration
randomNum = rand() % 3 //Set randomNum 0, 1, or 2

Comparisons:
userInput[0] == “r” || “R”
userInput[0] == “p” || “P”
userInput[0] == “s” || “S”

handUser[0] == handComp[0]
handUser[0] == rockU[0], paperU[0], scissorsU[0]
handComp[0] == rockU[0], paperU[0], scissorsU[0]

Constructors:
handUser[0, 1, 2, … 7] == “Rock” || “Paper” || “Scissors”
handComp[0, 1, 2, … 7] == “Rock” || “Paper” || “Scissors”

d. Design the logic of your program using pseudocode or flowcharts. See pseudocode syntax at the bottom of this document. Here is where you would use conditionals, loops, functions or array constructs (if applicable) and list the steps in transforming inputs into outputs. Walk through your logic steps with the test data from the assignment document.

```
START

DECLARE randomNum as INTEGER
DECLARE handComp[256] = "" as STRING
DECLARE handUser[256] = "" as STRING
DECLARE userInput[256] = "" as STRING
DECLARE rockL[256] = "rock" as STRING
DECLARE rockU[256] = "Rock" as STRING
DECLARE paperL[256] = "paper" as STRING
DECLARE paperU[256] = "Paper" as STRING
DECLARE scissorsL[256] = "scissors" as STRING
DECLARE scissorsU = "Scissors" as STRING

DISPLAY “Lets play Rock, Paper, Scissors!”

DISPLAY “Please type “Rock”, “Paper”, or “Scissors” to choose your hand.”
DISPLAY “Note: You can type “R”, “P”, or “S” too.”

DISPLAY “Please choose your Hand: ”
INPUT userInput
IF userInput[0] == “r” OR userInput[0] == “R”
   SET handUser = Rock
ELSE IF userInput[0] == “p” OR userInput[0] == “P”
   SET handUser = Paper
ELSE IF userInput[0] == “s” OR userInput[0] == “S”
   SET handUser = Scissors
ELSE
   DISPLAY userInStr + “is not a valid input.”
   DISPLAY “Please run the program again.”
   RETURN 0
DISPLAY “You have chosen ” + handUser

SET seedRandom = systemTime
SET randomNum = random (0 to 2 inclusive both)
IF randomNum == 0
   SET handComp = Rock
ELSE IF randomNum == 1
   SET handComp = Paper
ELSE
   SET handComp = Scissors
DISPLAY “The computer chose: “ + handComp

IF handUser == handComp
   DISPLAY “You both chose “ + handUser + “, it’s a tie!”
ELSE IF handUser == Rock AND handComp == Paper
   DISPLAY “You have chosen: “ + handUser + “!”
   DISPLAY “The computer chose: “ + handComp + “.”
   DISPLAY handComp + “ beats ” + handUser + “.”
```

```
    DISPLAY “The computer has won!”
ELSE IF handUser == Rock AND handComp == Scissors
    DISPLAY “You have chosen: “ + handUser + “!”
    DISPLAY “The computer chose: “ + handComp + “.”
    DISPLAY handUser + “ beats ” + handComp + “.”
    DISPLAY “Congratulations, you won!
ELSE IF handUser == Paper AND handComp == Scissors
    DISPLAY “You have chosen: “ + handUser + “!”
    DISPLAY “The computer chose: “ + handComp + “.”
    DISPLAY handComp + “ beats ” + handUser + “.”
    DISPLAY “The computer has won!”
ELSE IF handUser == Paper AND handComp == Rock
    DISPLAY “You have chosen: “ + handUser + “!”
    DISPLAY “The computer chose: “ + handComp + “.”
    DISPLAY handUser + “ beats ” + handComp + “.”
    DISPLAY “Congratulations, you won!
ELSE IF handUser == Scissors AND handComp == Rock
    DISPLAY “You have chosen: “ + handUser + “!”
    DISPLAY “The computer chose: “ + handComp + “.”
    DISPLAY handComp + “ beats ” + handUser + “.”
    DISPLAY “The computer has won!”
ELSE IF handUser == Scissors AND handComp == Paper
    DISPLAY “You have chosen: “ + handUser + “!”
    DISPLAY “The computer chose: “ + handComp + “.”
    DISPLAY handUser + “ beats ” + handComp + “.”
    DISPLAY “Congratulations, you won!
ELSE
    DISPLAY !Application error. Please alert the programmer!
    RETURN 0

DISPLAY “Thank you for playing!”

END
```

e. Include 2 Sample Program Runs for your program using your own set of data. This data set must be different from my Sample Runs in the Assignment document. This process is similar to Unit Testing and will help you test your program better.

Sample Program Run 1:

Lets play Rock, Paper, Scissors!

Please type “Rock”, “Paper”, or “Scissors” to choose your hand.
Note: You can type “R”, “P”, or “S” too.

Please choose your Hand: **Clips**

“**Clips**” is not a valid input.

Please choose your Hand: **Scisrsriuwn**

You have chosen **Scissors**!
The computer chose: Rock.

Rock beats Scissors.
The computer has won!

Thank you for playing.

Sample Program Run 2:

Lets play Rock, Paper, Scissors!

Please type “Rock”, “Paper”, or “Scissors” to choose your hand.
Note: You can type “R”, “P”, or “S” too.

Please choose your Hand: **Paper?**

You have chosen **Paper**!
The computer chose: Rock.

Paper beats Rock.
Congratulations, you won!

Thank you for playing!

Pseudocode Syntax

Think about each step in your algorithm as an action and use the verbs below:

To do this:	**Use this verb:**	**Example:**
Create a variable	DECLARE	`DECLARE integer num_dogs`
Print to the console window	DISPLAY	`DISPLAY “Hello!”`
Read input from the user into a variable	INPUT	`INPUT num_dogs`
Update the contents of a variable	SET	`SET num_dogs = num_dogs + 1`
Conditionals		

Use a single alternative conditional	IF *condition* THEN
	  statement
	  statement
		`IF num_dogs > 10 THEN`
		`  DISPLAY “That is a lot of dogs!”`

	END IF	`END IF`

Use a dual alternative conditional	IF *condition* THEN
	  statement
	  statement
	ELSE
	  statement
	  statement
	END IF
		`IF num_dogs > 10 THEN`
		`  DISPLAY "You have more than 10 dogs!"`
		`ELSE`
		`  DISPLAY "You have ten or fewer dogs!"`
		`END IF`

Use a switch/case statement	SELECT *variable or expression*
	  CASE *value_1:*
	  statement
	  statement
	  CASE *value_2:*
	  statement
	  statement
	  CASE *value_2:*
	  statement
	  statement
	  DEFAULT:
	  statement
	  statement
	END SELECT
		`SELECT num_dogs`
		`  CASE 0: DISPLAY "No dogs!"`
		`  CASE 1: DISPLAY "One dog.."`
		`  CASE 2: DISPLAY "Two dogs.."`
		`  CASE 3: DISPLAY "Three dogs.."`
		`  DEFAULT: DISPLAY "Lots of dogs!"`
		`END SELECT`

Loops		

Loop while a condition is true - the loop body will execute 0 or more times.	WHILE *condition*
	  statement
	  statement
	END WHILE
		`SET num_dogs = 1`
		`WHILE num_dogs < 10`
		`  DISPLAY num_dogs, " dogs!"`
		`  SET num_dogs = num_dogs + 1`
		`END WHILE`

Loop while a condition is true - the loop body will execute 1 or more times.	DO
	  statement
	  statement
	WHILE *condition*
		`SET num_dogs = 1`
		`DO`
		`  DISPLAY num_dogs, " dogs!"`
		`  SET num_dogs = num_dogs + 1`
		`WHILE num_dogs < 10`

Loop a specific number of times.	FOR *counter* = *start* TO *end*
	  statement
	  statement
	END FOR
		`FOR count = 1 TO 10`
		`  DISPLAY num_dogs, " dogs!"`
		`END FOR`

Functions		

Create a function	FUNCTION *return_type name (parameters)*
	  statement
	  statement
	END FUNCTION
		`FUNCTION Integer add(Integer num1, Integer num2)`
		`  DECLARE Integer sum`
		`  SET sum = num1 + num2`
		`  RETURN sum`
		`END FUNCTION`

Call a function	CALL *function_name*	`CALL add(2, 3)`
Return data from a function	RETURN *value*	`RETURN 2 + 3`