CSE 1321 CSE 1321 Test 1 A

Rutvik Dhira Marakana

TOTAL POINTS

91.875 / 95

QUESTION 1

1 Data Types 3.875 / 5

✓ **- 0.625 pts** Missed one primitive data type

- 1.25 pts Missed two of the primitive data types

- 1.875 pts Missed three of the primitive data types

- 2.5 pts Missed four of the primitive data types

- 3.125 pts Missed five of the primitive data types

- 3.75 pts Missed six of the primitive data types

- 4.375 pts Missed seven of the primitive data types

- 5 pts Missed eight of the primitive data types

✓ **- 0.5 pts** Listed String as primitive data type

- 0 pts Correct

QUESTION 2

2 Order of Operations 9 / 10

+ 1 pts Showed 1 step

+ 2 pts Showed 2 steps

+ 3 pts Showed 3 steps

+ 4 pts Showed 4 steps

+ 5 pts Showed 5 steps

+ 6 pts Showed 6 steps

+ 7 pts Showed 7 steps

✓ **+ 8 pts** Showed 8 steps

✓ **+ 1 pts** Incorrect final answer

+ 2 pts Correct Answer

+ 0 pts Unattempted

QUESTION 3

3 flag 5 / 5

✓ **+ 2 pts** flag value is correctly evaluated as "true"

✓ **+ 3 pts** output= is displayed

+ 0 pts blank or incorrect answer

QUESTION 4

4 Print A and B 10 / 10

✓ **+ 3 pts** A and B are declared

✓ **+ 2 pts** A and B are declared as integers

✓ **+ 3 pts** A + B assigned to K

✓ **+ 2 pts** A and B are initialized

QUESTION 5

5 Print V and W 10 / 10

✓ **+ 2 pts** V and W are declared

✓ **+ 2 pts** V and W are declared as integers

✓ **+ 3 pts** V*V assigned to W

✓ **+ 1 pts** V is initialized

✓ **+ 2 pts** PRINT "The value of W=" + W

+ 0 pts no or incorrect answer

QUESTION 6

6 Print input 7 / 10

✓ **+ 1 pts** explicit casting

✓ **+ 3 pts** declared variables

✓ **+ 3 pts** input assigned 123.45

+ 3 pts input/5 assigned to int_value

+ 0 pts no or incorrect answer

input <- input/5 not 123.45/5

QUESTION 7

7 Print flag 10 / 10

✓ **+ 3 pts** boolean flag <- false

✓ **+ 3 pts** flag <- 'A' < 'a'

✓ **+ 4 pts** print "flag = " , flag

+ 0 pts blank or incorrect answer

QUESTION 8

8 value is odd 15 / 15

✓ **+ 5 pts** Properly declared an integer variable called value and initialized it with input from the user.

✓ **+ 5 pts** Determines if the variable value is even or not using an if else statement.

✓ **+ 5 pts Prints proper message.**

+ 0 pts Incorrect logic.

+ 0 pts Pseudocode is vague and does not outline the necessary steps to determine if a number is even or odd.

+ 5 pts Determines if value is even or odd using a switch statement.

QUESTION 9

9 Boolean Age 5 / 5

✓ **- 0 pts Correct**

- 2.5 pts Split it into two expressions/forgot one check

- 1 pts Incorrect syntax/order

- 1 pts Put nothing, or OR or || instead of AND or &&

- 1 pts Reversed both signs/used equal to on both

- 0.5 pts Reversed one sign or used equal to one one

- 1 pts Did not include Age

- 5 pts Incomplete/Unattempted

QUESTION 10

10 case 2 / 0

+ 5 pts Correctly displays output (Number 3, Number 3, Number 4, Number 5)

✓ **+ 2 pts Displays only the results of 2 (Number 3)**

+ 0 pts Does not display correct output.

+ 3 pts Correctly displays the results of 2, 3, and 4, but does not include the output for OTHERS (Number 5)

QUESTION 11

11 if-else even odd 15 / 15

✓ **- 0 pts Correct/Good start**

- 2 pts Something missing/ not correct

- 7 pts Partially correct

- 15 pts Blank/ not answered

- 10 pts Incorrect attempt

- 5 pts Logic missed for finding Even, Odd, Zero and Negative

- 0 pts Click here to replace this description.

gradescope

MWF - 1pm

CSE 1321: Programming and Problem Solving I

Test 1

Student Name: Rutvik Mankana Date: 09/14/18

Q1: In module 2 we discussed 13 primitive data types used in programming. List 8 of them.

Answer:
1. int
2. long
3. short
4. char
5. bool
6. string
7. float
8. double

Q2: Using precedence rules show in detail how to evaluate the following expression and show the order of operations:

```
x = 100 - (40 + (200 / (10 - 5) * 2) / 4) + 50
```

Answer:

$x = 100 - (40 + (200 / (10 - 5) * 2) / 4) + 50$

$= 100 - (40 + (200 / 5 * 2) / 4) + 50$

$= 100 - (40 + (40 * 2) / 4) + 50$

$= 100 - (40 + 80 / 4) + 50$

$= 100 - (40 + 20) + 50$

$= 100 - 60 + 50 \quad = 100 - 110 = -10$

Q3: Evaluate the following pseudocode and write its exact output

```
flag ← false
output ← flag OR true
PRINT ("Output = " + output)
```

Answer:

Output = true

MWF-11

Q4: In module 2 we discussed variables declaration, initialization, and assignments. Write pseudocode for each of the following requirements. Do not write complete programs, just what is required for each part.

Part 1: Declare two integer variables (called A and B); initialize A to 20 and B to 40; assign the sum of A and B to K.

Answer:

```
int A, B
A ← 20
B ← 40
int K ← A + B
```

Part 2: Declare two integer variables (called V and W); initialize V to 50; assign V^2 to W; then printout the value of W with this label: **`The value of W =`**

Answer:

```
int V, W
V ← 50
W ← V * V
PRINT ("The value of W = " + W)
```

Part 3: Declare a variable of type double (call it `input`) and initialize it to 123.45; then change its value by dividing it by 5; finally assign the value of variable `input` to another integer variable (call it `int_value`) using explicit type casting. Make sure all variables are declared.

Answer:

```
double input
input ← 123.45
input ← 123.45/5.0
(int) int_value ← input
```

Part 4: Declare a boolean variable (call it `flag`) and initialize it to false; then assign the expression `'A' < 'a'` to variable `flag`; and finally print out the value of variable `flag` with this label **`Flag =`**

Answer:

```
bool flag ← false
flag ← 'A' < 'a'
PRINT ("Flag = " + flag)
```

MWF-1pm

Q5: Declare an integer variable (call it `value`); initialize the variable with an input from the user; check if the variable (`value`) is even or not; finally, print proper message such as "`value is even`" or "`value is odd`".

Answer:

```
int value
READ input_value from user for value
value ← input_value
```

continued on blank page

Q6: Write a boolean expression that evaluates to **true** if **age** is greater than 15 and less than 21.

Answer:

```
IF (age>15 && age<21)
    PRINT ("true")
ENDIF
```

Q7: Trace and print out the exact output of the following pseudocode.

```
n ← 2
CASE (n) OF
      1: PRINT("Number 1")
      2: PRINT("Number 3")
      3: PRINT("Number 3")
      4: PRINT("Number 4")
OTHERS:
         PRINT("Number 5")
ENDCASE
```

Answer:

Number 3

MWF-1p

Q8: Complete the following code using nested **IF-ELSE statements** to check whether a user input (integer value) is EVEN, ODD, ZERO, or NEGATIVE. Sample outputs of the code segment are as follows:

```
Entered value is:  34
This value is EVEN

Entered value is:  19
This value is ODD

Entered value is:  0
This value is ZERO

Entered value is:  -12
This value is NEGATIVE
```

Answer:

```
READ user_input for value
value ← user_input
PRINT("Entered value is: " + value)
IF (value % 2 == 0) THEN
    PRINT ("This value is EVEN")
ELSE IF (value == 0) THEN
    PRINT ("This value is ZERO")
ELSE IF (value < 0) THEN
    PRINT ("This value is NEGATIVE")
ELSE
    PRINT ("This value is ODD")
ENDIF
```

MWF - 1 pm

(Q.5) continued

```
IF (value % 2 ==0) THEN
    PRINT ("value is even")
ELSE
    PRINT ("value is odd")
ENDIF
```