**Computer Programming Quiz 2**

**Name:**

**Student Number:**

**Questions 1 to 4: Mark either True or False.**

[1] A char is stored internally as an array of String.

**Answer: True / False**

[2] Result of !(true && !(false || true)).

**Answer: True / False**

[3] Calls to methods returning boolean can be used as (logical) tests.

**Answer: True / False**

[4] while loop repeatedly executes its body until a logical test is true.

**Answer: True / False**

**Questions 5 to 8: Fill in the blank with a suitable word.**

[5] The chars in a String can be accessed using the \_\_\_\_\_\_ method.

**Answer:**

[6] == compares objects by \_\_\_\_\_\_, so it often gives false even when two Strings have the same sequence of letters.

**Answer:**

[7] A(n) \_\_\_\_\_\_\_ loop is a loop where the number of times its body repeats is known in advance.

**Answer:**

[8] When one needs to test whether two strings contain the same sequence of characters without considering cases, the \_\_\_\_\_\_\_ method could be used.

**Answer:**

**Questions 9 to 12: Write the expected result after invoking the following code excerpt.**

**If you think an excerpt cannot give a valid result, briefly state your reason.**

[9]

int i = 0, j = 10.0, k = 0;

while(i != j) {

k = k + i;

i = i + 1;

}

System.out.println(k);

**Answer:**

[10]

double cp\_gp = 3.33;

if(cp\_gp > 4.0)

System.out.print(“Satti”);

else if(cp\_gp > 3.5)

System.out.print(“Hagos”);

else if(cp\_gp > 3.3)

{

if(cp\_gp < 3.4)

System.out.print(“CP”);

else System.out.print(“BE”);

System.out.print(“ST”);

}

else if(cp\_gp > 3.0)

System.out.print(“Wonil”);

if(cp\_gp > 2.5)

System.out.print(“Yeonil);

System.out.println();

**Answer:**

[11] Assume this is the content of a file named cp11.java.

public class cp11 {

public static void main(String[] args) {

Random rand = new Random();

int quiz\_score = rand.nextInt(19);

if(quiz\_score < 20)

System.out.println(“Do your best!”);

}

}

**Answer:**

[12]

java.util.Scanner scan = new java.util.Scanner(System.in);

String first\_str = scan.nextLine();

String second\_str = “test”;

int score = scan.nextInt();

second\_str = scan.nextLine();

if(second\_str.startsWith(first\_str))

System.out.print(“One”);

else if(second\_str.constains(“m”))

System.out.print(“Two”);

else if(first\_str.contains(“m”))

System.out.print(“Three”);

else System.out.print(“Four”);

if(score == 20) System.out.print(“Five”);

System.out.println();

Assume an user entered *computer*, pressed an Enter key, entered *20*, pressed an Enter key, entered *programming* and pressed an Enter key.

**Answer:**

**This quiz continues on the next page.**

**Questions 13 to 15: Write a single statement per each bracket inside a method that determines whether a given number has an odd digit. You may assume n is a legit input (an integer greater than zero).**

public static boolean hasAnOddDigit(int n) {

do

{

if([13]) return true;

[14]

}

while([15])

return false;

}

**Answer for 13:**

**Answer for 14:**

**Answer for 15:**

**You may use this area if you do not have**

**enough space for your answer(s). Clearly indicate the corresponding question number.**

**=====================================**

**=====================================**

**Double check whether your name and student number are correctly written on the front page.**

**You may leave right after finishing and submitting your work.**

**Questions 16 to 20: In the lab session, we have done implementation of a simple card game. We will define a different scoring rule for this task. A card has its own id, consisting of two parts:**

**- First letter: C(0), D(2), H(4), S(6), K(8).**

**- Second letter/digit: A(12), 2(11), 3(10), 4(9), 5(8), 6(7), 8(5), 9(4), J(3), Q(2), K(1).**

**The score of a card is calculated by**

**(score of first letter) \* 10 +**

**(score of second letter/digit).**

**For instance, score of a card whose id is *SK* becomes 6 \* 10 + 1 = 61. Likewise, a card with *K9* contains a score value of 8 \* 10 + 4 = 84.**

**Complete the following method, using a single statement per bracket.**

public static int calculateScore(String id) {

int score = 0;

// Calculate the first score.

if(id.[16](“K”)) score = 80;

else if(id.[17](“S”)) score = 60;

…

// Assume the prior part is properly done.

// Calculate the second score.

// For A, J, Q.

if(id.[17](“A”)) score = score + 12;

else if(id.[17](“J”)) score = score + 3;

else if(id.[17](“Q”)) score = score + 2;

else

{

char second = id.[18](1);

// For K.

if(second == [19]) score = score + 1;

// For other digits.

else score = score + [20];

}

return score;

}

**Answer for 16:**

**Answer for 17:**

**Answer for 18:**

**Answer for 19:**

**Answer for 20:** (Hint: ASCII cod of ‘1’ is 49.)