CS108 Mock Test Time allowed 45 minutes

Part A Answer all questions 10 marks each

1. Complete the following code fragments:

(a) To print the numbers 10 20 30 40 50 … 200

```
int num;
for (int i= __ ;  _____ ; ___ ) {
        num = _______
        System.out.print(" " + num);
}
```

(b) To print the response based on the input marks/

grade HD	if marks in the range 80 – 100
grade D	if marks in the range 70 – 79
grade C	if marks in the range 60-69
grade P	if marks in the range 50 – 59
grade F	if marks in the range 0 - 49

Inavlid marks otherwise

```
System.out.println("Enter marks ");
int marks = console.readInt();
String response;

_____
_____
_____
_____
_____
_____
_____

System.out.println(response);
```

(c) To get a valid marks in the range 0 to 100 with a loop
print Error message “Marks too low. Re-enter” if it is less than 0
print Error message “Marks too high. Re-enter” if it is greater than 100

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

(d) Use nested for loops to print the table

1 2 3
2 4 6
3 6 9

for (_____
_____
_____
_____
_____
_____

(e) Complete the program segment below that will read values for a and b and print all numbers in the range a to b repeatedly. The program should terminate when b is less than a.

```
int a;
int b;

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
```

Part B (50 marks)

(a) Writing a Part class

Write a class named Part to model the motor vehicle parts maintained in a warehouse. It should have instance variables to store the ID, name, stock-level and reorder-level and unit-price. It should provide a constructor taking as arguments ID, name, stock-level and reorder-level and unit-price (price per item). It should provide accessors to get the ID, stock-level, reorder-level and unit-price. It should provide another accessor to print the part details (to System.out) including ID, name, stock-level, reorder-level and unit-price. It should provide two mutators one to withdraw and another to replenish both passing an integer (quantity) as argument. The withdraw() method should return false if the stock-level is lower than the quantity requested.

(b) Using the Part class

Using the class created above perform the following operations
// Create a Part array that can store up to 10 elements

// Construct 4 Part objects and Set the first 4 array references to refer to them
// First part "p1234" "nuts 1.5 cm" , 12000, 1000, 0.15
// Second part "p1235" "nuts 2.0 cm", 15000, 8000, 0.20
// Third part "p1236" "bolts 1.2 cm", 400, 300, 0.10
// Fourth part "p1237" "bolts 0.5 cm", 900, 400, 0.12

// Write a statement using a for loop to withdraw 500 items from all the existing parts
// If operation is successful print new stock-level otherwise print "insufficient stock"

// Print details of all items whose stock-level is lower than the reorder level

// Withdraw stock specifying ID.
System.out.println("Enter part ID")
String partID = console.readLine();
System.out.println("Enter Quantity")
int qty = console.readInt();
// Search through the array. If not found print error message otherwise perform operation