KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
Computer Science Department

CSC 111 Computer Programming - I	Lab Exam Spring 2025	Signature:

Name:	ID:	Lab Time:

Instructions:

1. Create a project with your full name as ***LabTime_FirstName_LastName_ID***
2. In the first 3 lines of your classes, write your ***full name, your ID, and your lab time*** as comments.
3. Use the specified variable and class names.
4. Name the exported compressed file as follows ***LabTime_FirstName_LastName_ID***
5. Submit your code to the Lab Exam section in blackboard.

Q1) Write a program that manages invoices for a parking lot. Your program should allow the user to add parking invoices, pay them, and print them. Here is the UML diagram:

Part1) Class: ParkingInvoice [7 Marks]

The class ParkingInvoice has the attributes:
type: String that represent type of parking ("short" or "long")
duration:int length of the parking (in hours)
paid: boolean variable that's true if the invoice is paid or false otherwise.

ParkingInvoice	Mark
- type : String - duration : int - paid : boolean	0.5
+ ParkingInvoice() + ParkingInvoice(String, int) + setType(String) : void + setDuration(int) : void + setPaid(boolean) : void + getDuration() : int + getPaid() : boolean + calcPrice() : int + print() : void	1 1 0.5 0.5 0.3 0.3 0.3 1.5 1

The methods are:

- **ParkingInvoice():** default constructor. Sets type to short, duration to 1 and paid to false.
- **ParkingInvoice(type, duration):** Constructor that sets type of parking and duration according to the argument provided by user. It also sets paid to false.
- **setType(type):** Assigns the value for type. Must be either the word short or long, if it's anything else, set it to short.
- **setDuration(duration)**: Assigns the value for duration. Must be a positive integer,
- otherwise set it to **1 Hour.**
- **setPaid(paid)**: Assigns the value for paid.
- **getDuration()**: Returns the value of duration.
- **getPaid()**: Returns the value of paid.
- **calcPrice():** Calculates and returns the price of the invoice based on its type and duration:
  - If type is **Short**: then parking rate is 10 SAR/hour.
    Example: For a **short** parking invoice that stayed for 7 hours the price is 7 X 10 = 70 SR
  - If type is **Long**: then parking rate is 5 SAR/hour.
    Example: For a **long** parking invoice that stayed for 20 hours, the price is 20 X 5 = 100 SR
- **print()**: prints out the invoice info in the following way (see sample run):
  - `Invoice [type: type, duration: duration hour(s), paid: yes/no, price: price SR.].`
  - `Example: Invoice [type: short, duration: 1 hour(s), paid: no, price: 10 SR.]`

Part2) Class: ParkingTest [4 Marks]

The class parking test is the class that you are going to use to test your program. It has the main method that

provides the user with a menu (once) that offers the following:

- add: prompts the user to enter the type & duration and creates a new unpaid invoice.

- pay: pays the current invoice. If there is no invoice print "`ERROR: no current invoice.`". If invoive was already paid print "`ERROR: already paid.`"

- print: prints the information of the current invoice. If there is no invoice print "`ERROR: no current invoice.`".

- exit: terminates the program and prints the sum of all **paid** invoices

Sample Run:

```
Welcome to the Parking Invoice System
-------------------------------------
add ==> add a new invoice.
pay ==> pay current invoice.
print ==> print current invoice.
exit ==> terminate the program.
-------------------------------------
Type your command.
>: pay↵
ERROR: no current invoice.
>: print↵
ERROR: no current invoice.
>: test↵
ERROR: incorrect command.
>: add↵
Enter the type and duration: short 1↵
>: print↵
Invoice [type: short, duration: 1 hour(s), paid: no, price: 10 SR.]
>: add↵
Enter the type and duration: long 10↵
>: print↵
Invoice [type: long, duration: 10 hour(s), paid: no, price: 50 SR.]
>: pay↵
Payment done.
>: print↵
Invoice [type: long, duration: 10 hour(s), paid: yes, price: 50 SR.]
>: pay↵
ERROR: already paid.
>: add↵
Enter the type and duration: short 45↵
>: print↵
Invoice [type: short, duration: 45 hour(s), paid: no, price: 450 SR.]
>: pay↵
Payment done.
>: pay↵
ERROR: already paid.
>: print↵
Invoice [type: short, duration: 45 hour(s), paid: yes, price: 450 SR.]
>: exit↵
Sum of paid invoices: 500 SR
Goodbye.
```