

APOLLO

B3 Object-oriented programming SL

Computer Science

First assessment 2027

2 questions | questions

Option A — Java

Answer **all** questions. All code that forms part of your responses must be written in Java.

3. A company employs a number of salespeople to sell its goods.

Details of the sales made by each employee for a year are stored in the `SalesPerson` object for that employee.

At the end of each month, the total sales for each employee are added to the array `sales` in the `SalesPerson` object. At the end of the year, the array will contain 12 entries, one for each month of the year.

Bonuses are paid to employees at the end of each year.

The `SalesPerson` class is partially shown:

```
public class SalesPerson
{
    static int bonus1 = ... ;           // value to be added
    static int bonus2 = ... ;           // value to be added
    private String id;                  // unique id for each employee
    double[] sales = new double[12];    // sales figures for each month in a year

    public SalesPerson(String id)
    {
        this.id = id;
    }

    public String getId() // returns employee's id
    {
        return this.id;
    }

    public double calculateTotal() // returns the total sales figure for
the year
    {
        // code missing
    }

    public double calculateHighest() // returns highest monthly total in
the year
    {
        // code missing
    }

    public void addSales(...) // adds the sales for a month to the sales array
    {
        // code missing
    }
}
```

(Option A continues on the following page)

— 3 —

(Option A, question 3 continued)

The driver code can be used to instantiate new `SalesPerson` objects. For example:

```
SalesPerson employee1 = new SalesPerson ("A6457098");
```

- (a) List the operations that would be carried out by the constructor when this line of code is run.

[4]

An object can have different types of attributes.

- (b) Outline **one** difference in the way in which the `bonus1` and `id` attributes are applied in the `SalesPerson` class.

[2]

The variable `id` has been identified as being a private attribute of the class.

- (c) Outline **two** consequences of declaring an attribute as private.

[2]

At the end of each month, the `SalesPerson` method, `addSales()`, adds the total sales for that month to the sales array for each employee.

The parameters passed to this method are the total sales for that month and the number representing that month, where 0 corresponds to January.

For example:

```
employee1.addSales(30000,2); // this adds 30000 to the sales array for  
the month of March for employee1.
```

- (d) Construct the method `addSales()`.

[2]

The `SalesPerson` method `calculateHighest()` returns the highest monthly sales figure by that employee during the year.

- (e) Construct the method `calculateHighest()`.

[5]

(Option A continues on page 11)**Turn over**

Blank page

— . . —

(Option A, question 3 continued)

The company has 10 employees. An array, `employee`, of size 10 holds `SalesPerson` objects, one for each employee.

End-of-year bonuses, shown in the `SalesPerson` class, are calculated for each employee as follows:

- `bonus1` is paid if the total sales for the year are more than 300 000.
- `bonus2` is paid if the sales for any one month are more than 20 000.

The driver code includes the method `calculateBonus()`.

For each employee, this method:

- calculates the bonuses
- outputs their employee ID and their total bonus.

(f) Construct the method `calculateBonus()`.

Use any of the methods listed in the `SalesPerson` class.

[5]

End of Option A

Turn over

Option B — Python

Answer **all** questions. All code that forms part of your responses must be written in Python.

6. A company employs a number of salespeople to sell its goods.

Details of the sales made by each employee for a year are stored in the `SalesPerson` object for that employee.

At the end of each month, the total sales for each employee are added to the list `self._sales` in the `SalesPerson` object. At the end of the year, the list will contain 12 entries, one for each month of the year.

Bonuses are paid to employees at the end of each year.

The `SalesPerson` class is partially shown:

```
class SalesPerson():

    bonus1 = ...          # value to be added
    bonus2 = ...          # value to be added

    def __init__(self, id):
        self.__id = id      # employee id
        self.__sales = []   # holds the monthly sales figures

    def get_id(self):        # returns employee's id
        return self.__id

    def calculate_total(...): # returns the total sales figure for the year
        # code missing

    def calculate_highest(...): # returns highest monthly total in the year
        # code missing

    def add_sales(...):      # adds the sales for a month to the sales list
        # code missing
```

(Option B continues on the following page)

(Option B, question 6 continued)

The driver code can be used to instantiate new `SalesPerson` objects. For example:

```
employee1 = SalesPerson ("A6457098")
```

- (a) List the operations that would be carried out by the constructor when this line of code is run. [4]

An object can have different types of attributes.

- (b) Outline **one** difference in the way in which the `bonus1` and `self._id` attributes are applied in the `SalesPerson` class. [2]

The underscore character in `self._id = id` identifies the variable as being a private attribute of the class.

- (c) Outline **two** consequences of declaring an attribute as private. [2]

At the end of each month, the `SalesPerson` method, `add_sales()`, adds the total sales for that month to the sales list for each employee.

For example:

```
employee1.add_sales(30000) # this adds 30000 to the sales list
for employee1.
```

- (d) Construct the method `add_sales()`. [2]

The `SalesPerson` method `calculate_highest()` returns the highest monthly sales figure by that employee during the year.

- (e) Construct the method `calculate_highest()`.

The `max()` function must not be used. [5]

(Option B continues on page 21)

Turn over

Blank page

— 41 —

(Option B, question 6 continued)

The company has 10 employees. A list, `employee`, of size 10 holds `SalesPerson` objects, one for each employee.

End-of-year bonuses, shown in the `SalesPerson` class, are calculated for each employee as follows:

- `bonus1` is paid if the total sales for the year are more than 300 000.
- `bonus2` is paid if the sales for any one month are more than 20 000.

The driver code includes the function `calculate_bonus()`.

For each employee, this function:

- calculates the bonuses
- outputs their employee ID and their total bonus.

(f) Construct the function `def calculate_bonus()`.

Use any of the methods listed in the `SalesPerson` class.

[5]**End of Option B**
