Grade 11 Computing

Unit 1 · Objects and Classes

WORKSHEET

Full marks 33

Name _____ Class _____ Roll no. ____ Date _____ Marks _____ / 33

1 Key words [4]

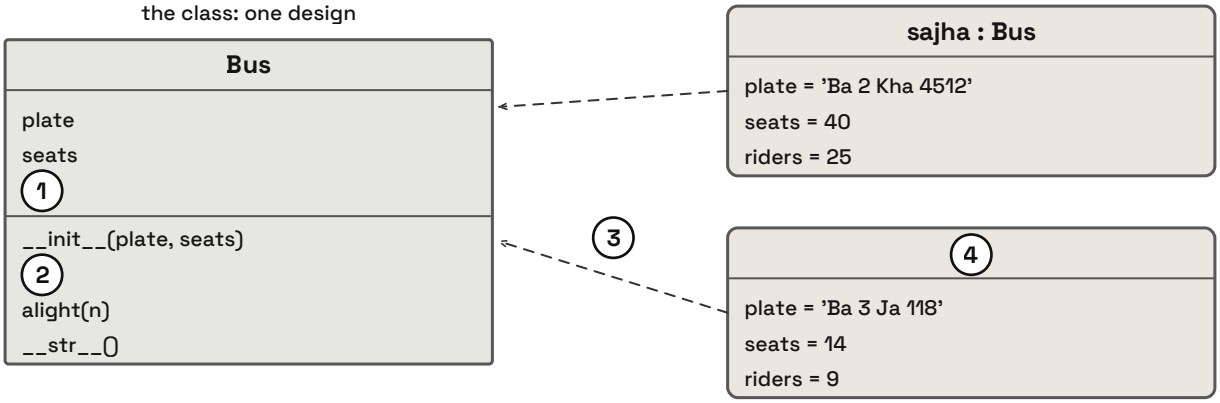
Fill in the blanks with words from the box. Two words in the box are not needed. Each blank carries 0.5 mark.

attribute	class	composition	constructor	encapsulation	inheritance	method	module	object	recursion

A _____ is a design, and each thing made from it is an _____. A value an object keeps about itself is an _____; a function that belongs to the class is a _____. Python calls `__init__`, the _____, each time a new object is made. Keeping an object's data behind its methods, so the methods can check every change, is _____. A child class that gets every method of its parent uses _____; a class that holds objects of another class, as a route holds its stops, uses _____.

2 Label the diagram [2]

The diagram shows the class Bus and two objects made from it. Write the word or words that belong in place of 1 to 4. Each carries 0.5 mark.



1. _____ 2. _____

3. _____ 4. _____

3 Multiple choice [5]

Choose the correct option for each of the following questions.

i. Inside a method, what does **self** refer to? [1]

a. the class **b.** the object the method was called on

c. the first object ever made from the class **d.** the method itself

ii. `a = Bus("Ba 2 Kha 4512", 40)` and `b = Bus("Ba 2 Kha 4512", 40)`. Then `a.board(10)`. What is `b.riders`? [1]

a. 0

b. 10

c. 20

d. an error, because two buses have the same plate

iii. Inside `__init__`, a student writes `marks = []` instead of `self.marks = []`. What happens when a method later uses `self.marks`? [1]

a. It works: Python adds the self. by itself.

b. The list is shared by every object.

c. AttributeError: the object has no attribute marks, because `marks = []` only made a local variable of `__init__`.

d. NameError: name 'marks' is not defined.

iv. Which pair is best joined by **inheritance** (the first a child class of the second)? [1]

a. Library and Book

b. Microbus and Vehicle

c. Route and Stop

d. Student and School

v. Every savings account at a cooperative earns the same interest rate, so the class has `rate = 0.06` written in it, outside every method. The cooperative raises the rate for every account to 7%. Which line does that? [1]

a. `sita.rate = 0.07`

b. `self.rate = 0.07`, in a method of the class

c. `Savings.rate = 0.07`

d. `rate = 0.07`, in a method of the class

4 What does it print? [5]

Write exactly what each program prints, in the box under it.

a. [2]

```
1  class Ticket:
2      sold = 0
3
4      def __init__(self, name):
5          self.name = name
6          Ticket.sold = Ticket.sold + 1
7
8  a = Ticket("Asha")
9  b = Ticket("Bimal")
10 a.sold = 50
11 c = Ticket("Gita")
12 print(a.sold, b.sold, Ticket.sold)
```

OUTPUT

b. [3]

```
class Piece:
    def __init__(self, at):
        self.at = at
    def name(self):
        return "piece"
    def show(self):
        return self.name() + " at " + str(self.at)

class Goat(Piece):
    def name(self):
        return "goat"
class Tiger(Piece):
    def show(self):
        return "TIGER " + super().show()

for p in [Goat(3), Tiger(0), Piece(7)]:
    print(p.show())
```

OUTPUT

5 Find and fix the bug [3]

Read the program, then answer.

Each student should have their own list of marks, but the program prints `Asha [72, 58]`, with Bimal's mark in Asha's list.

```
class Student:
    marks = []
    def __init__(self, name):
        self.name = name
    def add(self, mark):
        self.marks.append(mark)

asha = Student("Asha")
bimal = Student("Bimal")
asha.add(72)
bimal.add(58)
print(asha.name, asha.marks)
```

i. Why does Bimal's mark end up in Asha's list?
ii. Write the lines of the class that fix it.
iii. What does the correct program print?

6 Trace the program [4]

Fill in the trace table: a row for each call of a method, then what the program prints. The first row shows the values before the calls.

```
1  class Account:
2      def __init__(self):
3          self._balance = 0
4          self.refused = 0
5      def deposit(self, amount):
6          if amount > 0:
7              self._balance = self._balance + amount
8      def withdraw(self, amount):
9          if amount <= self._balance:
10             self._balance = self._balance - amount
11         else:
12             self.refused = self.refused + 1
13
14 sita = Account()
15 hari = sita
16 for amount in [500, -100, 300]:
17     sita.deposit(amount)
18 hari.withdraw(1000)
19 hari.withdraw(250)
20 print(sita._balance, sita.refused)
```

Call	_balance	refused	Output
(before)	0	0	

7 Short answers [4]

Write short answers to the following questions.

a. Should each pair be joined by inheritance or by composition? Give a reason for each: i. an Ambulance and a Vehicle; ii. a School and its Classrooms. [2]

b. In the Bagh-Chal program of **What does it print?**, the loop calls `p.show()` on three different kinds of object. What is this called, and why is it useful? [2]

8 Write a program [4]

Write the program in Python on the lines. Any program that gives the right result gets the marks.

Write a class **Hostel** with attributes `name`, `beds` (the number of beds) and `residents` (a list of names, empty at first), and three methods: `admit(person)` adds the person only if a bed is free, and returns True if they were admitted and False if not; `leave(person)` takes the person off the list; and `__str__` gives text such as `Girls' hostel: 1 of 2 beds taken`. Then make a hostel with 2 beds, admit Asha, Gita and Mina, let Asha leave, and print the hostel.

9 Think [2]

Think, then answer.

The cooperative's Account keeps its balance as a number in `_balance`, and other code reads it only through a method, `balance()`. The manager now wants every deposit and withdrawal kept in a passbook. Why does using `balance()`, and never `_balance`, save work? Python doesn't stop anyone writing `sita._balance = 1000000`. What, then, is the underscore for?

10 Activity: Mistakes with classes (not marked)

At a computer, open the unit's page (the QR code) and find the activity **Mistakes with classes.** For each program, find the line Python names, the kind of error and the fix, then fill in the table.

Program	Line Python names	Kind of error	The line that fixes it
1			
2			
3			