**Practical 2**

**PART 1**

class Item {

protected int location;

protected String description;

public Item(int location, String description) {

this.location = location;

this.description = description;

}

public int getLocation() {

return location;

}

public void setLocation(int location) {

this.location = location;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

}

class Monster extends Item {

public Monster(int location, String description) {

super(location, description); // Call the superclass constructor to initialize inherited variables

}

}

public class Main {

public static void main(String[] args) {

Item item = new Item(1, "A shiny gem");

System.out.println("Item description: " + item.getDescription());

System.out.println("Item location: " + item.getLocation());

Monster monster = new Monster(2, "Ferocious dragon");

System.out.println("Monster description: " + monster.getDescription());

System.out.println("Monster location: " + monster.getLocation());

}

}

**PART 2**

1. b) super
2. ?
3. b) private
4. b) Packages
5. c) import pkg.\*
6. c) charAt()
7. d) length()

**PART 03**

1)Real-world objects contain state and behavior.

2)A software object's state is stored in instance variables.

3)A software object's behavior is exposed through methods.

4)Hiding internal data from the outside world, and accessing it only through publicly exposed methods is known as data encapsulation.

5)A blueprint for a software object is called a class.

6)Common behavior can be defined in a base class and inherited into a derived class using the extends keyword.

7)A collection of methods with no implementation is called an interface.

8)A namespace that organizes classes and interfaces by functionality is called a package.

9)The term API stands for Application Programming Interface.