**Java lab sheet-01**

**1)**

**.**package com.mycompany.generalassignment;

//1.

public class Item

{

private int location;

private String description;

//2.

public Item(int l,String d

//3.

{

this.location=l;

this.description=d;

}

//4.

public void setlocation(int newlocation){

this.location=newlocation;

}

public int getlocation(){

return location;

}

public void setdescrition(String newdescription){

this.description=newdescription;

}

public String getdescription(){

return description;

}

//display

public void display()

{

System.out.println("Initial Location is:"+location);

System.out.println("Initial Description is:"+description);

}

}

//5.

package com.mycompany.generalassignment;

public class Monster extends Item

{//6.

public Monster(int location,String description){

super(location,description);

}

}

//7.

package com.mycompany.generalassignment;

public class GeneralAssignment

{

public static void main (String args[])

{

Monster m1=new Monster (12345,"ABC");

m1.display();

m1.setlocation(56789);

m1.setdescrition("EFG");

System.out.println("Update Location is:"+m1.getlocation());

System.out.println("Update Description is:"+m1.getdescription());

}

}

**PART 2**

1). b 3).b 4).b 5).c 6).c 7).d

**PART 3**

1. Real-world objects contain **attributes** and **behaviours.**

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

3. A software object's behaviour 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 behaviour can be defined in a **superclass** and inherited into a **subclass** 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**.