# Practical 02

Student ID: **27607**

Student Name: **PGWickramasuriya**

## 01.

```
public class Item {     protected
int location;     protected String
description;


    // Constructor     public Item(int location,
String description) {         this.location =
location;         this.description = description;

    }


    // Getters and Setters
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;

    }
}
```

02.

```
public class Item {     protected
int location;     protected String
description;


    // Constructor     public Item(int location,
String description) {         this.location =
location;         this.description = description;

    }


    // Getters and Setters (same as in the previous code)
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;

   }
}
```

03.

```
public class Item {     protected
int location;     protected String
description;


   // Constructor     public Item(int location,
String description) {        this.location =
location;        this.description = description;

   }
```

```
    // Getters and Setters (same as in the previous code)
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;

    }

}
```

## 04.

```
public class Item {     protected
int location;     protected String
description;



    // Constructor     public Item(int location,
String description) {         this.location =
location;         this.description = description;
```

```
    }


    // Getters     public int
getLocation() {         return
location;

    }


    public String getDescription() {
return description;

    }

    // Setters     public void
setLocation(int location) {
this.location = location;

    }


    public void setDescription(String description) {
this.description = description;

    }
}
```

## 05.

```
public class Monster extends Item {
    // Additional properties and methods specific to the Monster class can be added here.
```

```
   // Constructor for the Monster class     public
Monster(int location, String description) {
super(location, description);

      // Additional initialization specific to the Monster class can be added here.
   }

   // Additional methods specific to the Monster class can be added here.
}
```

06.

```
public class Monster extends Item {
   // Additional properties and methods specific to the Monster class can be added here.


   // Constructor for the Monster class     public
Monster(int location, String description) {
super(location, description);

      // Additional initialization specific to the Monster class can be added here.
   }


   // Constructor that takes an integer and a String argument
public Monster(int location, String description) {
super(location, description);

      // Additional initialization specific to the Monster class can be added here.
   }

   // Additional methods specific to the Monster class can be added here.
}
```

## 07.

```
public class Monster extends Item {
   // Additional properties and methods specific to the Monster class can be added here.


   // Constructor for the Monster class     public
Monster(int location, String description) {
super(location, description);

       // Additional initialization specific to the Monster class can be added here.
   }

   // Additional methods specific to the Monster class can be added here.
}
```

## Part 2

Part02

1.b.supper
2.b. private

3.b. package

4.c. import package

5.c. charAt()

6.d. length()

## Part03

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 parent class 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