PART 01:

1. Create a new class called ‘Item’ with two protected instance variables (private variables), an integer variable called ‘location’, and a String variable called ‘description’.

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

2. Add a constructor method for the Item class that takes an integer and a String as arguments (in that order).

```
public class Item
{
    protected int location;
    protected String description;
    public Item(int location, String description)
 {
        this.location = location;
        this.description = description;
          }
}
```

3. The constructor should assign the value of these parameters to the corresponding instance variables.
4. Add getter and setter methods for the location and description variables.

```
public 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;
    }
}
```

5. Add another class called Monster and make the Monster class a sub-class of the Item class.

```
public class Monster extends Item
 {


 }
```

6. Add a constructor method to the Monster class that takes an integer and a String argument just like the Item class constructor.

```
public class Monster extends Item
 {
    public Monster(int location, String description)
         {
        super(location, description);
         }
 }
```

7. Use these arguments to call the Item super class constructor from within the Monster class constructor so that the instance variables in the superclass are instantiated correctly.

```
public 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;
    }
}

public class Monster extends Item {
    public Monster(int location, String description) {
        super(location, description);

    }


}
```

PART 02

1. Which of these keywords is used to refer to member of base class from a sub class?
   a) upper b) super c) this d) None of the mentioned

3. The modifier which specifies that the member can only be accessed in its own class is
   a) public b) private c) protected d) none

4. Which of these is a mechanism for naming and visibility control of a class and its content?
   a) Object b) Packages
   c) Interfaces d) None of the Mentioned.

5. Which of the following is correct way of importing an entire package 'pkg'?
   a) import pkg. b) Import pkg.
   c) import pkg.* d) Import pkg.*

6. Which of these method of class String is used to extract a single character from a String object?
   a) CHARAT() b) charat()
   c) charAt() d) CharAt()

7. Which of these method of class String is used to obtain length of String object?
   a) get() b) Sizeof()
   c) lengthof() d) length()

PART 03: Fill in the blanks using appropriate term.

1. Real-world objects contain attributes 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 child 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.