

IS 667 Spring 2024

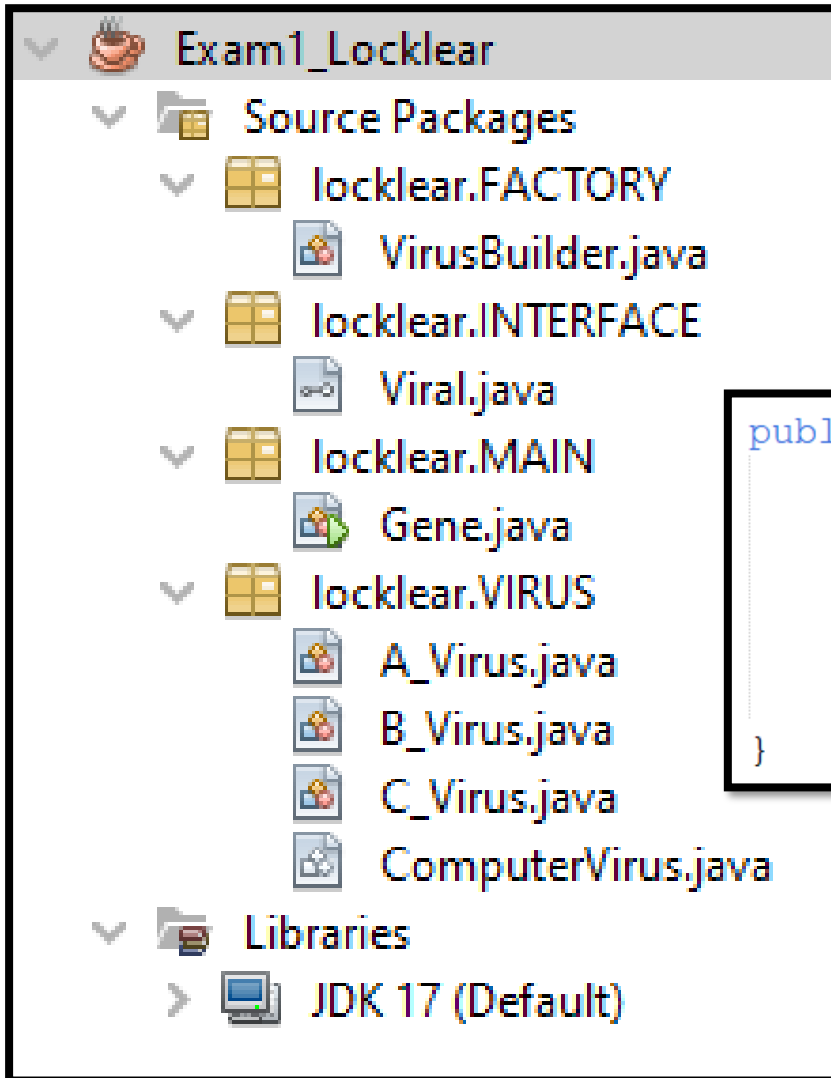
Dynamic Web Application Development

EXAM 1: Fundamental Java OOP

Professor HG Locklear

General

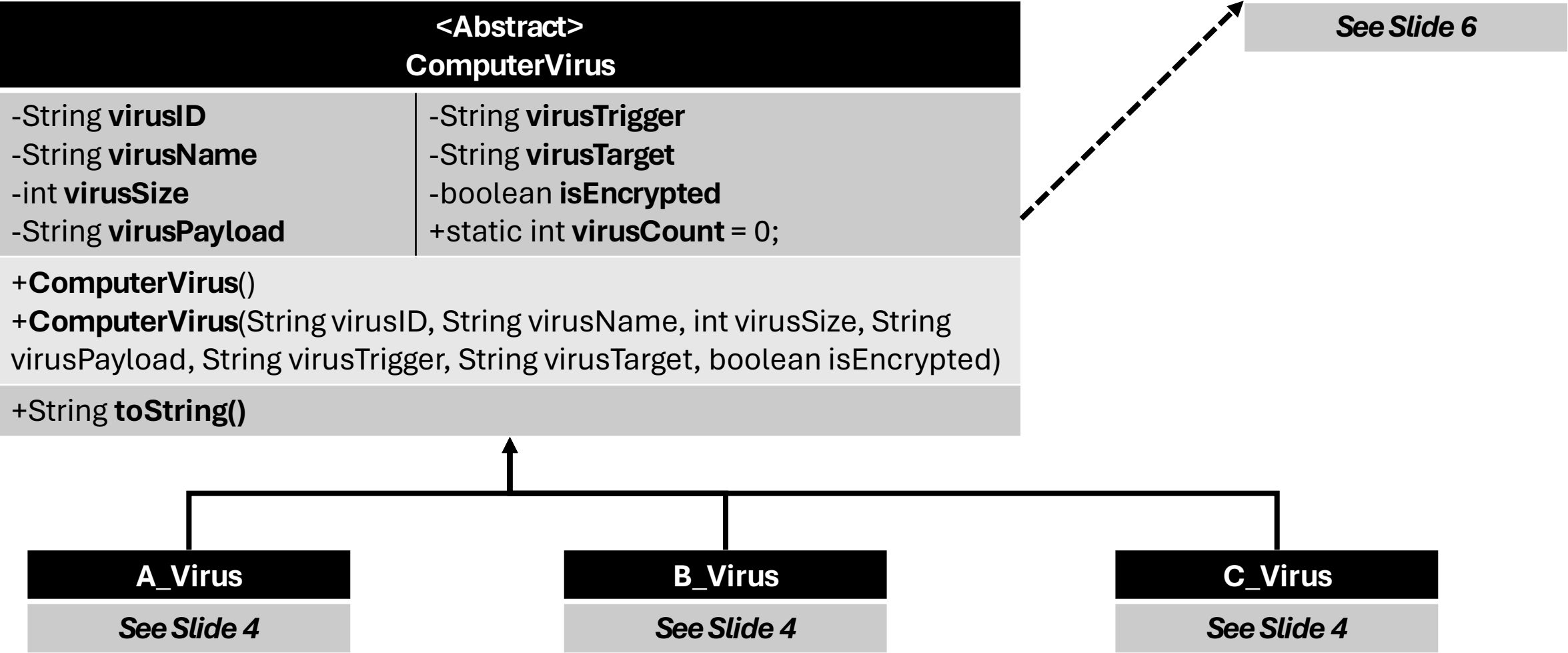
Create your **program structure** EXACTLY as shown replace my name with yours where appropriate



```
public class Gene {  
  
    public static void main(String[] args) {  
        ComputerVirus[] viruses = VirusBuilder.buildViruses(1,1,1);  
        VirusBuilder.displayViruses(viruses);  
    }  
}
```

Create your **main method** EXACTLY as shown replace my name with yours where appropriate

ComputerVirus Class



Both constructors must increment the **virusCount** by 1 at subclass instantiation

ComputerVirus Subclasses

A_Virus

-String **ACode** = "VIRUS_ALPHA"
-String virusName = "A" + virusID
-String virusSize = 4

+**A_Virus**(String virusID, String virusPayload, String virusTrigger, String virusTarget, boolean isEncrypted)

+void **displayMe()**

B_Virus

-String **BCode** = "VIRUS_BETA"
-String virusName = "B" + virusID
-String virusSize = 6

+**B_Virus**(String virusID, String virusPayload, String virusTrigger, String virusTarget, boolean isEncrypted)

+void **displayMe()**

C_Virus

-String **CCode** = "VIRUS_GAMMA"
-String virusName = "C" + virusID
-String virusSize = 8

+**C_Virus**(String virusID, String virusPayload, String virusTrigger, String virusTarget, boolean isEncrypted)

+void **displayMe()**

All subclasses **must use** superclass constructor.

ComputerVirus and Subclasses Methods

Class	Method Name	Input	Processing	Output
ComputerVirus	toString	None	Returns a String representation of the ComputerVirus object in the format shown below.	String
A_Virus	displayMe	None	Prints to the console a String representation of the object in the format shown below. This method <u>must utilize</u> the toString method of the ComputerVirus class.	Prints to Console
B_Virus				
C_Virus				

```
Virus ID: C1A
Virus Name: AC1A
Virus Size: 4
Virus Payload: Worm
Virus Trigger: Macro Run
Virus Target: Linux
Encrypted: YES
```

toString

```
***B Virus***
BCode: VIRUS_BETA
Virus ID: A2B
Virus Name: BA2B
Virus Size: 6
Virus Payload: Worm
Virus Trigger: Macro Run
Virus Target: Linux
Encrypted: NO
```

displayMe

Viral Interface

<<Viral>>

```
+static String[ ] ltrs = {"A","B","C","D","E","F"};
+static String[ ] digits = {"1","2","3","4","5","6"};
+static String[ ] payload = {"Worm", "Trojan Horse", "Spyware"};
+static String[ ] trigger = {"File Download", "Macro Run"};
+static String[ ] target = {"Windows", "Linux"};
+abstract void displayMe();
+static String generateID()
```

Method Name	Input	Processing	Output
displayMe	See Slide 5		
generateID	None	Returns a <u>random</u> 3-digit alphanumeric sequence consisting of a letter a number and a letter. <i>This method <u>must utilize the ltrs and digits</u> arrays</i>	String

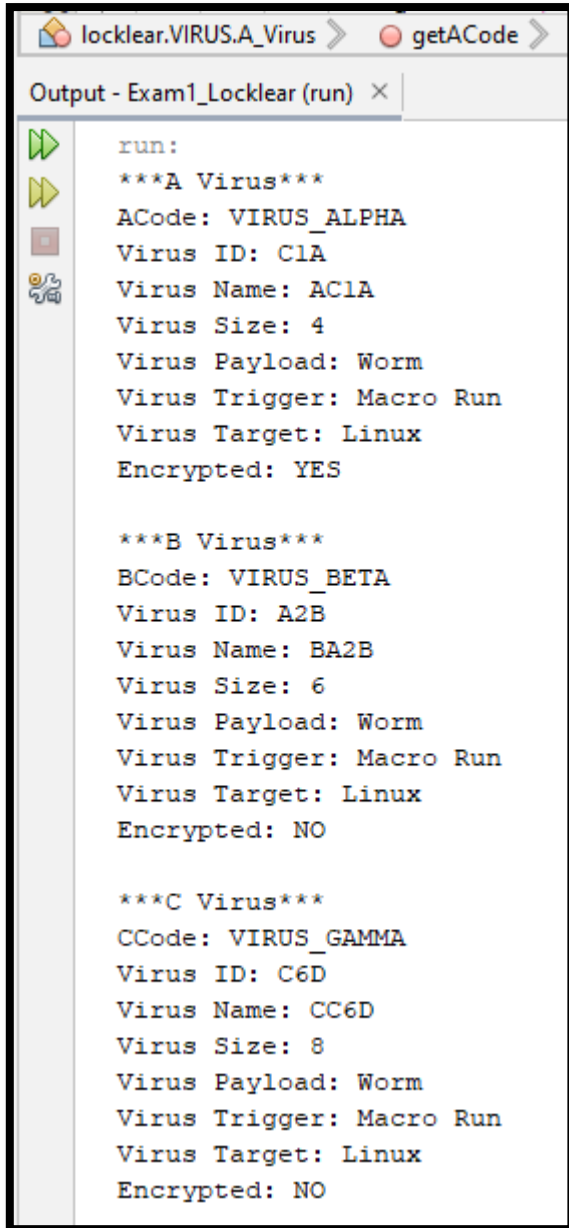
VirusBuilder Class

VirusBuilder

```
+static ComputerVirus[ ] buildViruses(int A, int B, int C)  
+static void displayViruses(ComputerVirus[ ] v)
```

Method Name	Input	Processing	Output
buildViruses	int A/B/C ...the number of A/B/C Viruses to build	Creates the specified number of each virus type and stores them in a 1D array and <u>returns</u> the array. Each constructor argument for a virus is <u>randomly generated</u> utilizing only the appropriate static variables contained in the Viral interface	1D Array
displayViruses	Computer[] v ...1D array of ComputerVirus objects	Calls the displayMe method for each ComputerVirus object in the argument 1D array	Prints to Console

Expected Output



```
run:
***A Virus***
ACode: VIRUS_ALPHA
Virus ID: C1A
Virus Name: AC1A
Virus Size: 4
Virus Payload: Worm
Virus Trigger: Macro Run
Virus Target: Linux
Encrypted: YES

***B Virus***
BCode: VIRUS_BETA
Virus ID: A2B
Virus Name: BA2B
Virus Size: 6
Virus Payload: Worm
Virus Trigger: Macro Run
Virus Target: Linux
Encrypted: NO

***C Virus***
CCode: VIRUS_GAMMA
Virus ID: C6D
Virus Name: CC6D
Virus Size: 8
Virus Payload: Worm
Virus Trigger: Macro Run
Virus Target: Linux
Encrypted: NO
```

Your attribute values may be different (randomly generated) but the output format must be EXACTLY as shown