**Introduction:**

A screenshot of a social media post

Description generated with very high confidence

**Prerequisites:**

1. **class.forname(“ <input>”):**

* prototype: public static Class forName(String class\_name) throws ClassNotFoundException
* It will load class byte code to the memory without creating object for the respective class.
* Class c = Class.forName("com.require.forNameClass");
* JVM will check for path in current directory, java libraries and location provided by classPath variable, if not found JVM will throw classCastException.
* So here JVM loads forNameClass byte code to the memory and then JVM every takes every details of that class and that information is stores as java.lang.Class object in heap memory.then reference variable is stored in Class variable c.

1. **newInstance():**

* prototyoe: public Object newInstance() throws InstantiationException, IllegalAccessException
* It able to create an object for the loaded class.
* Object obj = c.newInstance();
* JVM will check for loaded class and in that it checks for Zero argument public constructor and creates object using it.

**Dynamic Input approaches:** If we provide data to application at run time

1. Buffered Reader:

A screenshot of a cell phone

Description generated with very high confidence

Here System class has a static variable “in” which holds InputStream object. This inputStream object is connected to command Promt which takes the input as bytes.To convert bytes to chars we use InputStremReader. To improve the performance we use BufferedReader which gives output as Strings.

BufferedReader method has two methods to read the data.

* read() : It will read single character in the form of its ASCII value
  + public int read()
* readLine() : It will read a line of text in the form of String.
  + Public String readLine();

1. **Scanner:**

Java,util.scanner it is able to read the primitive data directly.

* To read primitive data as dynamic input.
  + Public xxx nextXxx();
* To read String as a dynamic input.
  + Public String next() // read single line delimited by space
  + Public String nextLine(); // read a line of text.

1. **Console:**

To overcome the limitations of BufferedReader such as data security and code optimization we go for console.

Console con = System.console();

* To display the message and to read the dynamic input in the form of string.
  + Public String readLine(String message)
* To display a message and to read password data in the form char[].
  + Public char[] readPassword(String message)

**File Input Stream and Output Stream:**

1. **FileInputStream:**

**A screenshot of a cell phone

Description generated with very high confidence**

1. **FileOutputStream :**

**A screenshot of a social media post

Description generated with very high confidence**