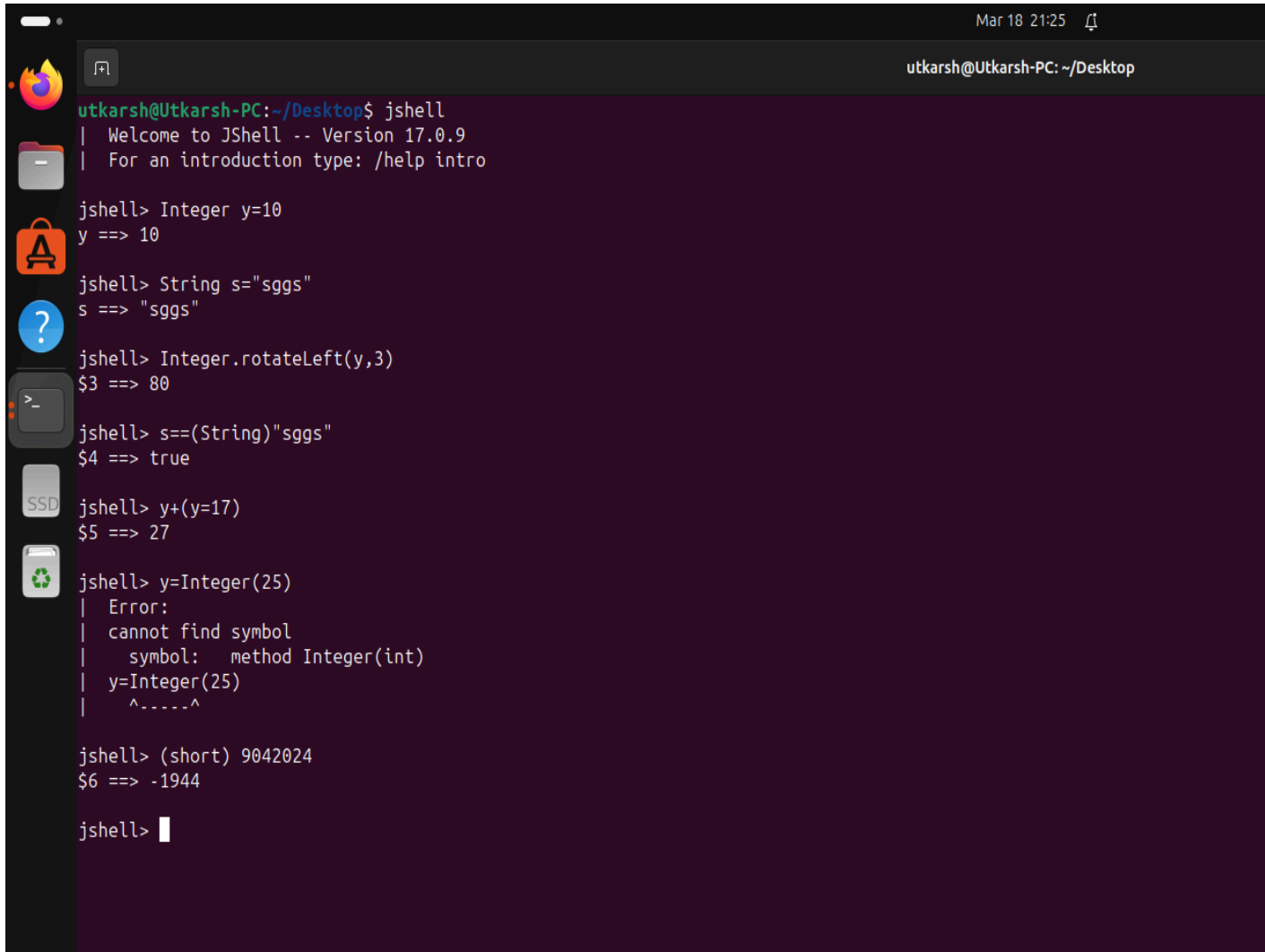


MIDTERM

Q1)Ans:



```
utkarsh@Utkarsh-PC: ~/Desktop
utkarsh@Utkarsh-PC:~/Desktop$ jshell
| Welcome to JShell -- Version 17.0.9
| For an introduction type: /help intro

jshell> Integer y=10
y ==> 10

jshell> String s="sggs"
s ==> "sggs"

jshell> Integer.rotateLeft(y,3)
$3 ==> 80

jshell> s==(String)"sggs"
$4 ==> true

jshell> y+(y=17)
$5 ==> 27

jshell> y=Integer(25)
| Error:
| cannot find symbol
|   symbol:   method Integer(int)
|   y=Integer(25)
|   ^.....^

jshell> (short) 9042024
$6 ==> -1944

jshell> 
```

Q2)

a)Ans:

Incorrect class name, Incorrect file permission, Incorrect compilation error, missing.class file.

c)Ans:

```
SSD
jshell> String s="utkarsh"
s ==> "utkarsh"

jshell> s[3]
| Error:
| array required, but java.lang.String found
| s[3]
| ^.-^

jshell>
```

b)Ans:

```
utkarsh@Utkarsh-PC:~/Desktop$ jshell
| Welcome to JShell -- Version 17.0.9
| For an introduction type: /help intro

jshell> String s="utkarsh
| Error:
| unclosed string literal
| String s="utkarsh
| ^

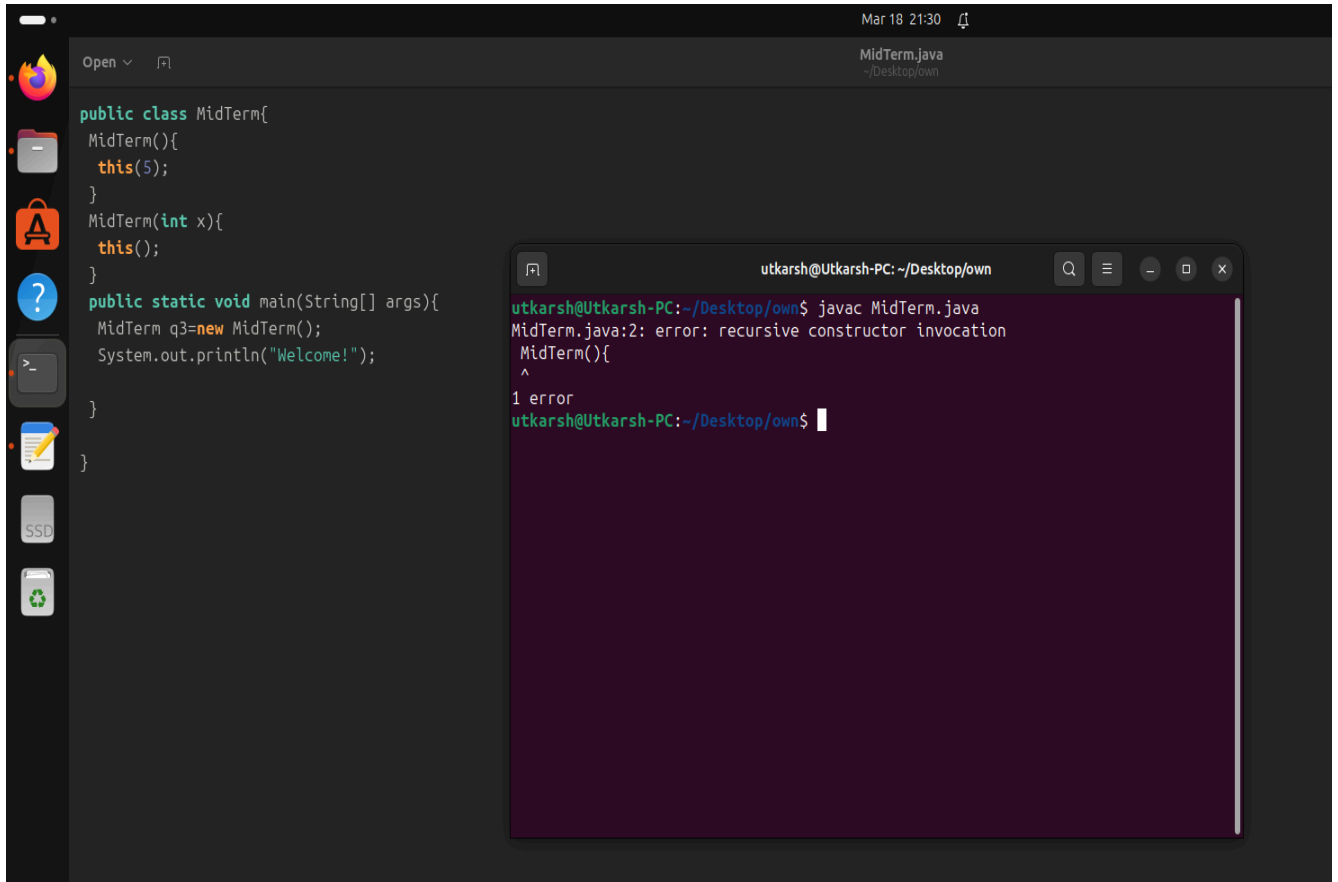
jshell> █
```

Q3)Ans:

Javac // latest version // preview file.name

Q4)

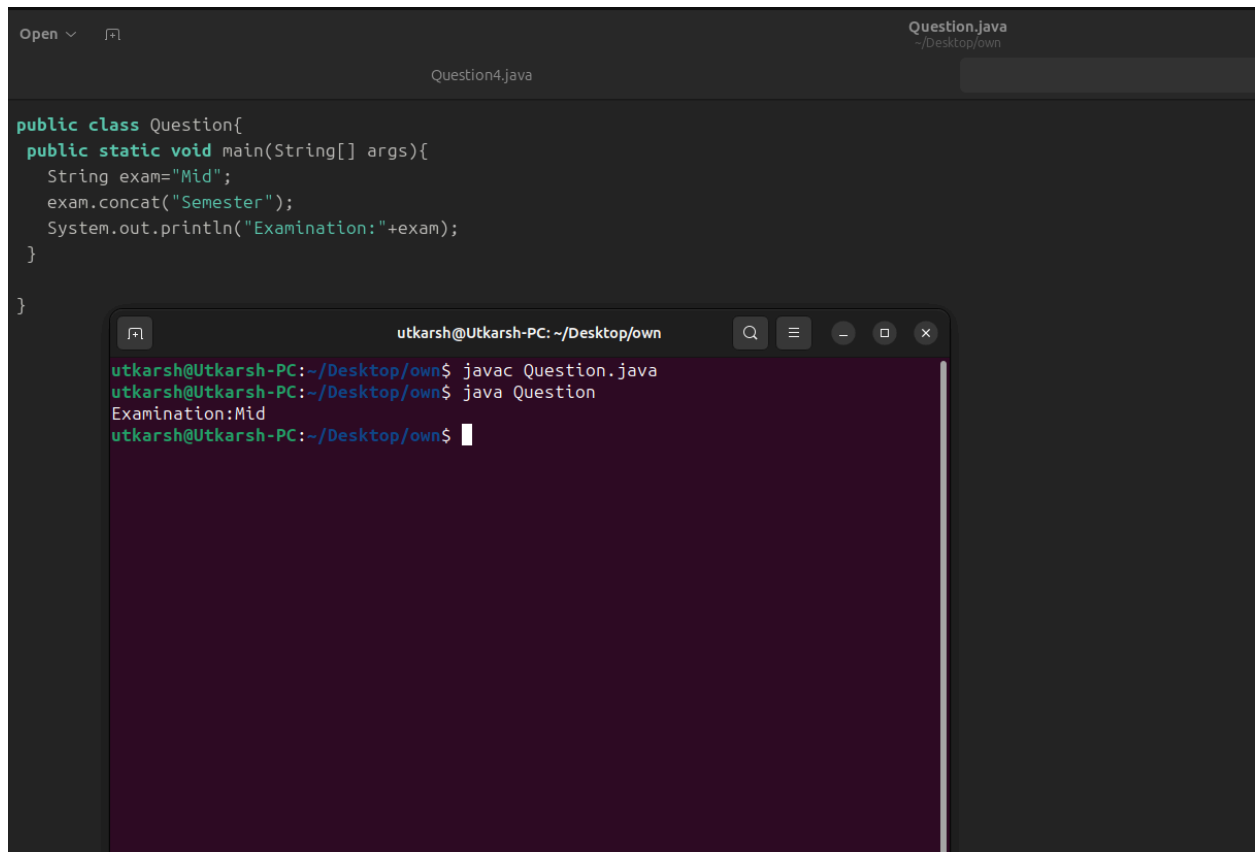
a)Ans:



```
public class MidTerm{
    MidTerm(){
        this(5);
    }
    MidTerm(int x){
        this();
    }
    public static void main(String[] args){
        MidTerm q3=new MidTerm();
        System.out.println("Welcome!");
    }
}
```

```
utkarsh@Utkarsh-PC: ~/Desktop/own
utkarsh@Utkarsh-PC:~/Desktop/own$ javac MidTerm.java
MidTerm.java:2: error: recursive constructor invocation
    MidTerm(){
    ^
1 error
utkarsh@Utkarsh-PC:~/Desktop/own$
```

b)Ans:

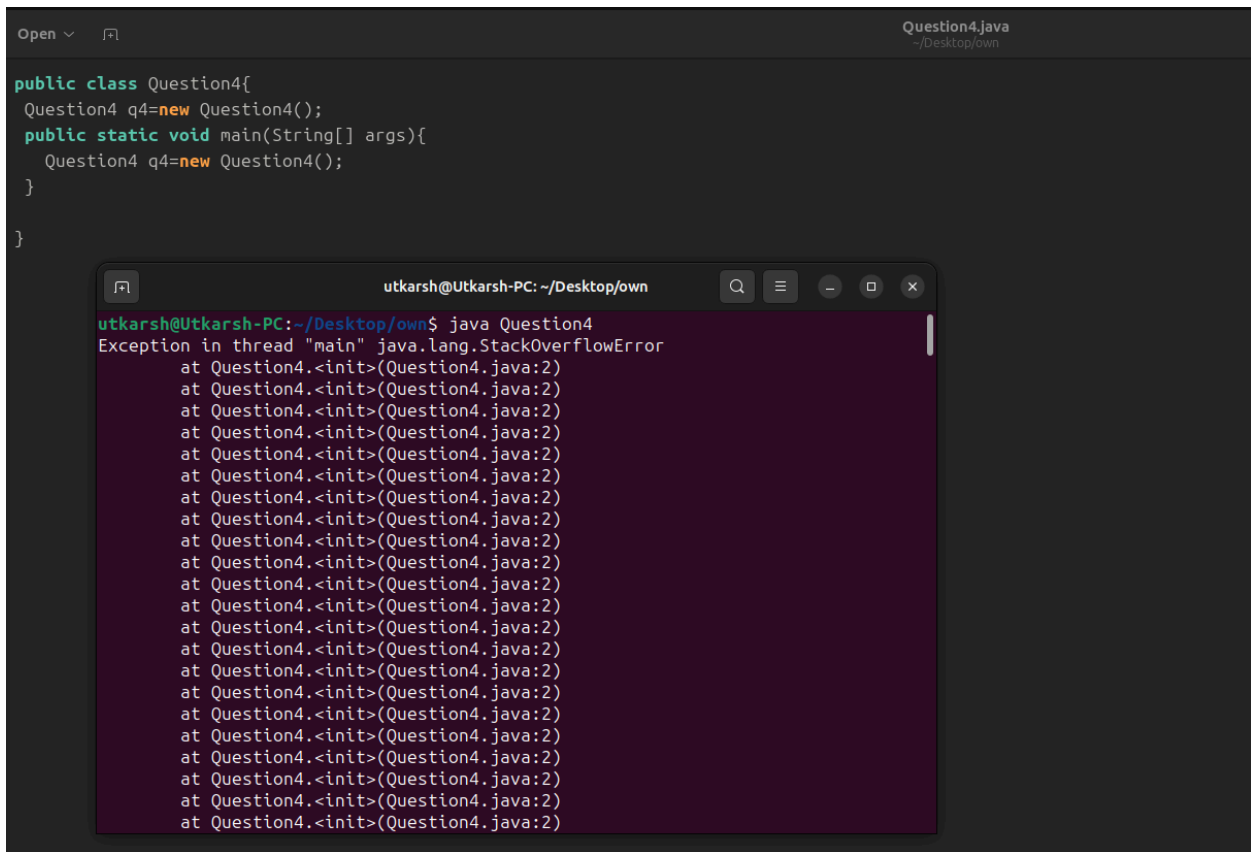


The image shows a screenshot of a code editor and a terminal window. The code editor displays a Java class named `Question` with a `main` method. The `main` method initializes a `String` variable `exam` with the value `"Mid"`, concatenates `"Semester"` to it, and prints the result using `System.out.println`. The terminal window shows the execution of the program, displaying the output `Examination:Mid`.

```
public class Question{  
    public static void main(String[] args){  
        String exam="Mid";  
        exam.concat("Semester");  
        System.out.println("Examination:"+exam);  
    }  
}
```

```
utkarsh@Utkarsh-PC: ~/Desktop/own  
utkarsh@Utkarsh-PC:~/Desktop/own$ javac Question.java  
utkarsh@Utkarsh-PC:~/Desktop/own$ java Question  
Examination:Mid  
utkarsh@Utkarsh-PC:~/Desktop/own$
```

c)Ans:



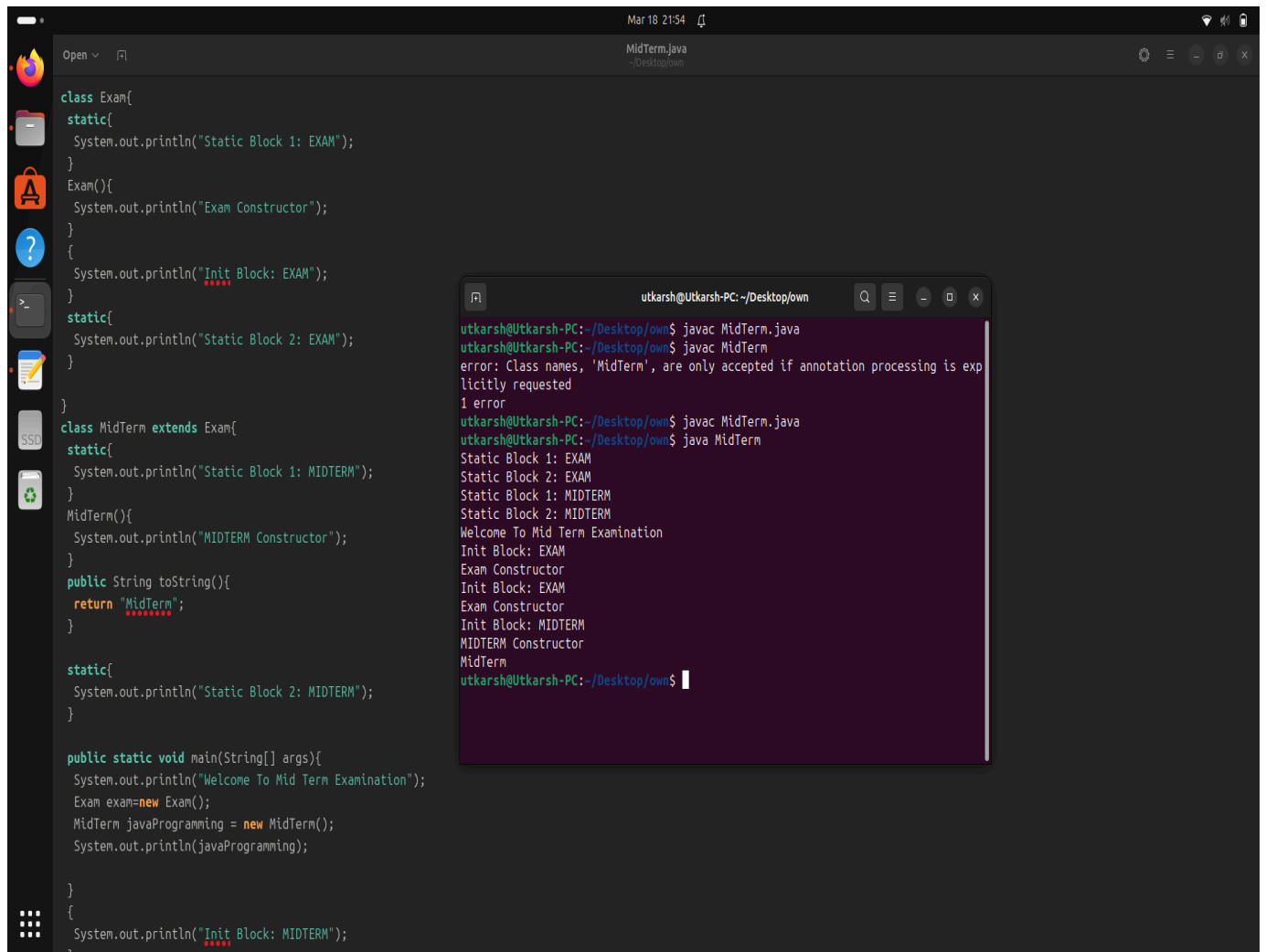
The image shows a screenshot of an IDE with a dark theme. At the top, a tab labeled 'Question4.java' is open, showing the following code:

```
public class Question4{
    Question4 q4=new Question4();
    public static void main(String[] args){
        Question4 q4=new Question4();
    }
}
```

Below the code editor, a terminal window is open, displaying the output of running the program. The terminal shows a stack overflow error:

```
utkarsh@Utkarsh-PC: ~/Desktop/own$ java Question4
Exception in thread "main" java.lang.StackOverflowError
    at Question4.<init>(Question4.java:2)
    at Question4.<init>(Question4.java:2)
    at Question4.<init>(Question4.java:2)
    at Question4.<init>(Question4.java:2)
    at Question4.<init>(Question4.java:2)
    at Question4.<init>(Question4.java:2)
    at Question4.<init>(Question4.java:2)
    at Question4.<init>(Question4.java:2)
    at Question4.<init>(Question4.java:2)
    at Question4.<init>(Question4.java:2)
    at Question4.<init>(Question4.java:2)
    at Question4.<init>(Question4.java:2)
    at Question4.<init>(Question4.java:2)
    at Question4.<init>(Question4.java:2)
    at Question4.<init>(Question4.java:2)
    at Question4.<init>(Question4.java:2)
    at Question4.<init>(Question4.java:2)
    at Question4.<init>(Question4.java:2)
    at Question4.<init>(Question4.java:2)
    at Question4.<init>(Question4.java:2)
```

d)Ans:



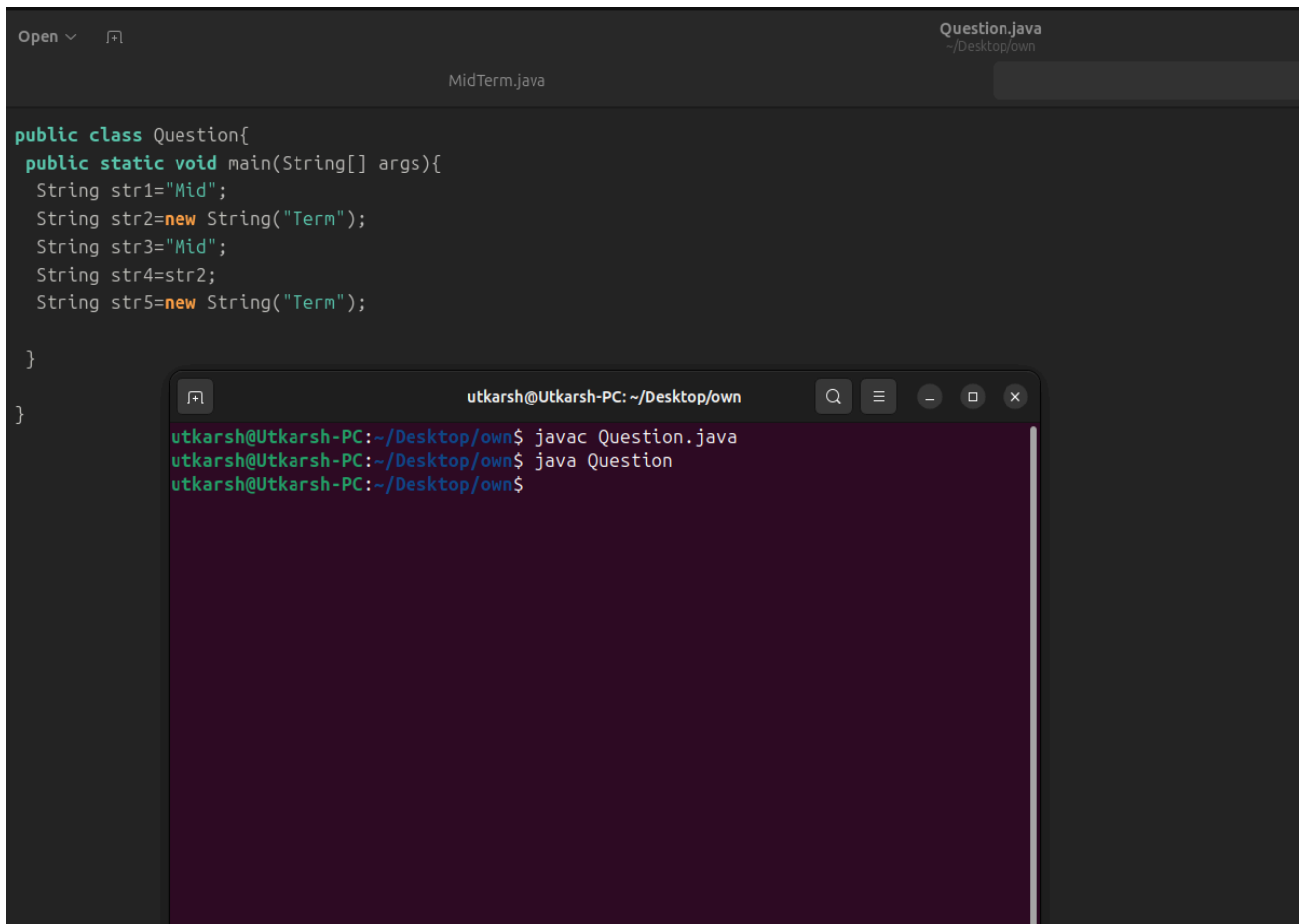
```
class Exam{
    static{
        System.out.println("Static Block 1: EXAM");
    }
    Exam(){
        System.out.println("Exam Constructor");
    }
    {
        System.out.println("Init Block: EXAM");
    }
    static{
        System.out.println("Static Block 2: EXAM");
    }
}

class MidTerm extends Exam{
    static{
        System.out.println("Static Block 1: MIDTERM");
    }
    MidTerm(){
        System.out.println("MIDTERM Constructor");
    }
    public String toString(){
        return "MidTerm";
    }
    static{
        System.out.println("Static Block 2: MIDTERM");
    }

    public static void main(String[] args){
        System.out.println("Welcome To Mid Term Examination");
        Exam exam=new Exam();
        MidTerm javaProgramming = new MidTerm();
        System.out.println(javaProgramming);
    }
    {
        System.out.println("Init Block: MIDTERM");
    }
}
```

```
utkarsh@Utkarsh-PC: ~/Desktop/own
utkarsh@Utkarsh-PC:~/Desktop/own$ javac MidTerm.java
utkarsh@Utkarsh-PC:~/Desktop/own$ javac MidTerm
error: Class names, 'MidTerm', are only accepted if annotation processing is explicitly requested
1 error
utkarsh@Utkarsh-PC:~/Desktop/own$ javac MidTerm.java
utkarsh@Utkarsh-PC:~/Desktop/own$ java MidTerm
Static Block 1: EXAM
Static Block 2: EXAM
Static Block 1: MIDTERM
Static Block 2: MIDTERM
Welcome To Mid Term Examination
Init Block: EXAM
Exam Constructor
Init Block: EXAM
Exam Constructor
Init Block: MIDTERM
MIDTERM Constructor
MidTerm
utkarsh@Utkarsh-PC:~/Desktop/own$
```

Q5)Ans:



The image shows a screenshot of a code editor with a dark theme. At the top, there are two tabs: 'MidTerm.java' and 'Question.java'. The 'Question.java' tab is active, displaying the following Java code:

```
public class Question{  
    public static void main(String[] args){  
        String str1="Mid";  
        String str2=new String("Term");  
        String str3="Mid";  
        String str4=str2;  
        String str5=new String("Term");  
    }  
}
```

Below the code editor, a terminal window is open, showing the command prompt for 'utkarsh@Utkarsh-PC: ~/Desktop/own'. The terminal contains the following commands and their outputs:

```
utkarsh@Utkarsh-PC:~/Desktop/own$ javac Question.java  
utkarsh@Utkarsh-PC:~/Desktop/own$ java Question  
utkarsh@Utkarsh-PC:~/Desktop/own$
```