

Name Nikhil Gupta
Roll No. 20051523
Batch CSE-06
Branch CSE
Subject Web Tech

Assignment 8

Ques 1 Dynamic Method Dispatch

```
class First
{
    void print() {
        System.out.println("First Start Class");
    }
}

class Derived extends First {
    void print() {
        System.out.println("Dynamic Method Dispatch");
    }
}

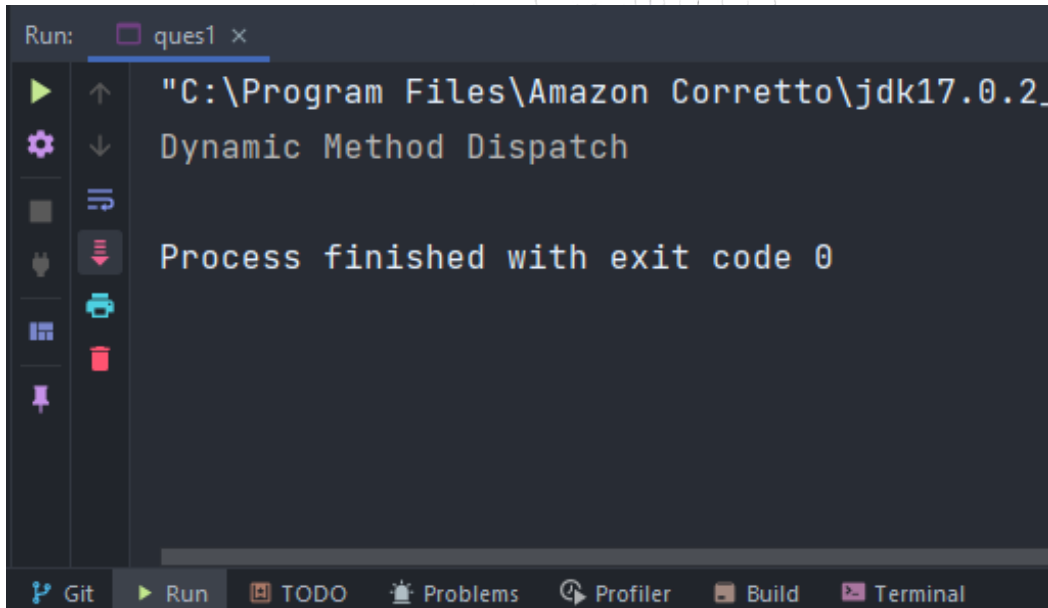
public class ques1 {
    public static void main(String[] args) {
        First fir;
    }
}
```

```
fix = new derived();
```

```
fix.print();
```

```
}
```

```
}
```



The screenshot shows the 'Run' console of an IDE. The title bar says 'Run: ques1 x'. The console output shows the Java runtime path: '"C:\Program Files\Amazon Corretto\jdk17.0.2_...' followed by 'Dynamic Method Dispatch' and 'Process finished with exit code 0'. The IDE's status bar at the bottom includes icons for Git, Run, TODO, Problems, Profiler, Build, and Terminal.

Ques 2

```
import java.util.Scanner;
```

```
public class ques2 {
```

```
    public static void main(String[] args) {
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        Person arr[] = new Person[5];
```

```
        for (int i = 0; i < 2; i++) {
```

```
            arr[i] = new Person();
```

```
System.out.println("Enter Your Name: ");
```

```
arr[i].name = sc.nextLine();
```

```
System.out.println("Enter Your Aadhar No. : ");
```

```
arr[i].aadharNo = sc.nextInt();
```

```
System.out.println("Enter Your Account No. : ");
```

```
arr[i].accNo = sc.nextInt();
```

```
System.out.println("Enter Your Bank Balance: ");
```

```
arr[i].balance = sc.nextInt();
```

```
}
```

```
System.out.println("Name \t Aadhar No. \t  
Account No. \t Account Bal ");
```

```
PLines(80);
```

```
for (int i=0 ; i<2 ; i++) {
```

```
    arr[i].display();
```

```
}
```

```
}
```

```
}
```

```
public static void pLines(int Lines)
```

```
{
```

```
    for(int i=0; i < lines; i++)
```

```
    {
```

```
        System.out.print("- ");
```

```
    }
```

```
    System.out.println(" ");
```

```
}
```

```
}
```

```
}
```

```
class Account {
```

```
    int accNo;
```

```
    int balance;
```

```
    void display() {
```

```
        System.out.println("Account No.: " + accNo);
```

```
        System.out.println("Balance: " + balance);
```

```
    }
```

```
}
```

```
}
```

class Person extends Account {

String name;

int aadharNo;

Person() {

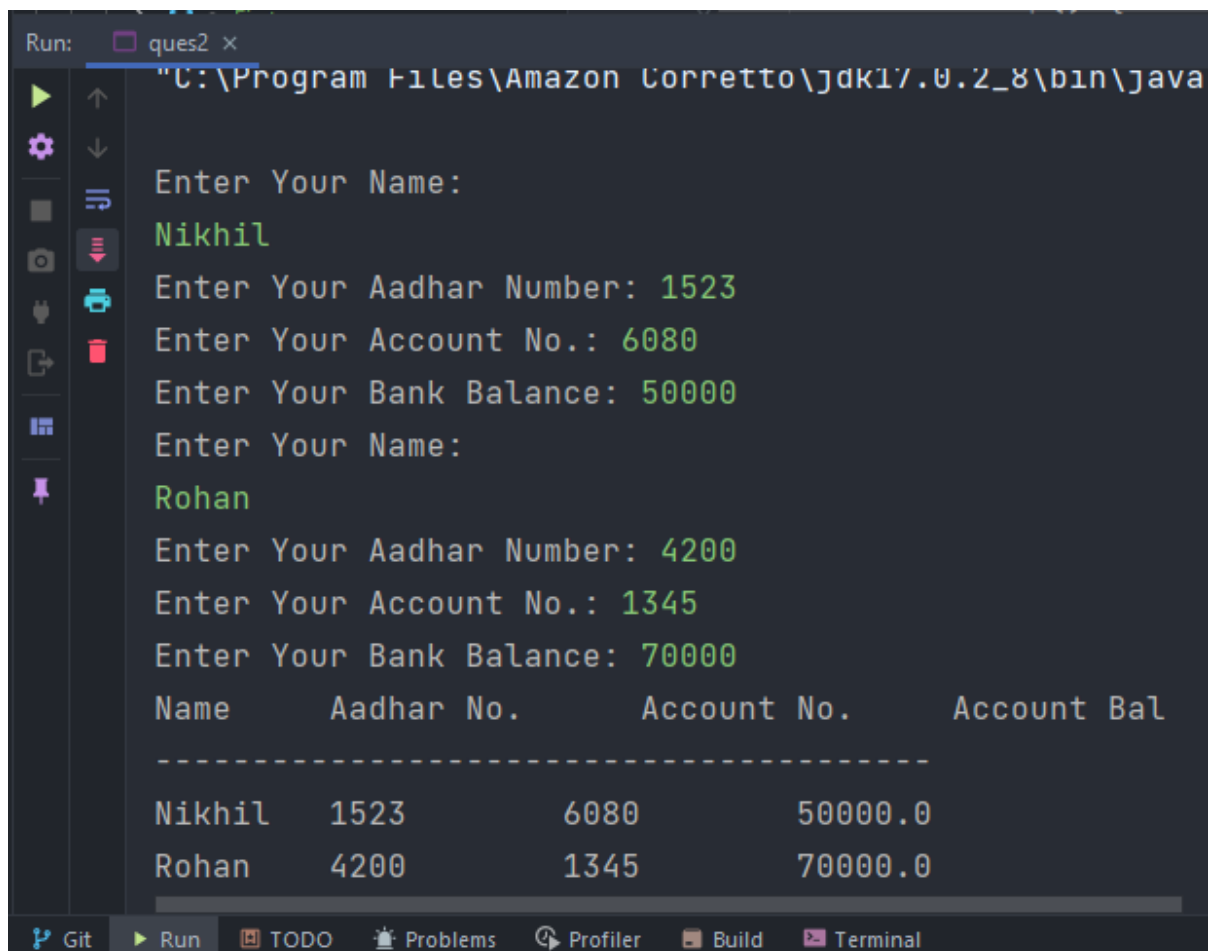
}

void display() {

System.out.println(Name + "\t" + AadharNo +
"\t" + accNo + "\t" + balance);

}

}



```
Run: ques2 x
"C:\Program Files\Amazon Corretto\jdk17.0.2_8\bin\java

Enter Your Name:
Nikhil
Enter Your Aadhar Number: 1523
Enter Your Account No.: 6080
Enter Your Bank Balance: 50000
Enter Your Name:
Rohan
Enter Your Aadhar Number: 4200
Enter Your Account No.: 1345
Enter Your Bank Balance: 70000
Name      Aadhar No.      Account No.      Account Bal
-----
Nikhil    1523            6080            50000.0
Rohan     4200            1345            70000.0
```

Ques 3

```
class Animal {
```

```
    Animal ()
```

```
    {
```

```
        System.out.println("I am an animal");
```

```
    }
```

```
    Animal(String type) {
```

```
        System.out.println("Type: " + type);
```

```
    }
```

```
}
```

```
class Dog extends Animal {
```

```
    Dog() {
```

```
        super("Animal");
```

```
        System.out.println("I am a Dog");
```

```
    }
```

```
}
```

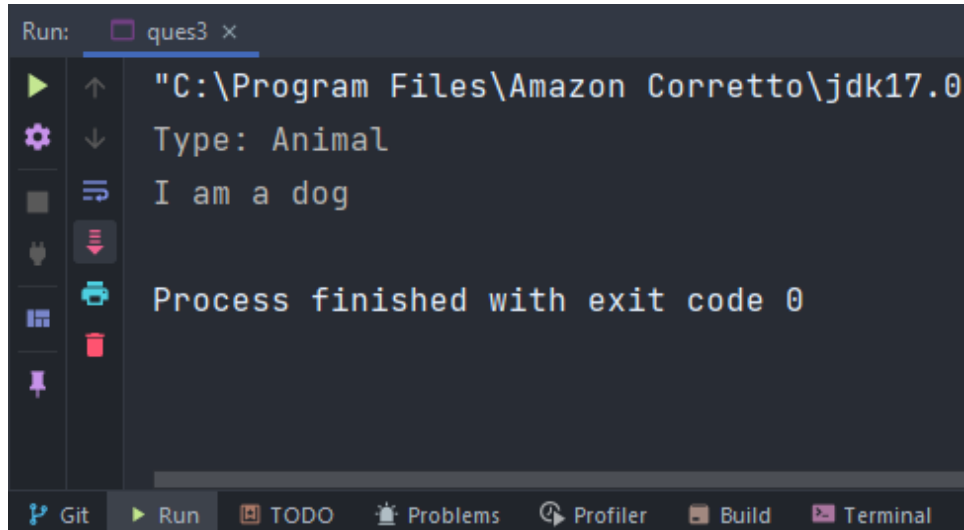
```
public class ques3 {
```

```
    public static void main(String[] args) {
```

```
Dog dog1 = new Dog();
```

```
}
```

```
}
```



The screenshot shows the 'Run' console of an IDE. The title bar says 'Run: ques3 x'. The output text is as follows:

```
"C:\Program Files\Amazon Corretto\jdk17.0
Type: Animal
I am a dog
Process finished with exit code 0
```

At the bottom, there is a toolbar with icons for Git, Run, TODO, Problems, Profiler, Build, and Terminal.

Ques 4

```
class A {
```

```
    int PC() {
```

```
        return 0;
```

```
    }
```

```
}
```

```
class B extends A {
```

```
    public static void int PC() {
```

```
        return 5;
```

```
    }
```

```
}
```



```

class C extends A {
    protected int PC() {
        return 6;
    }
}

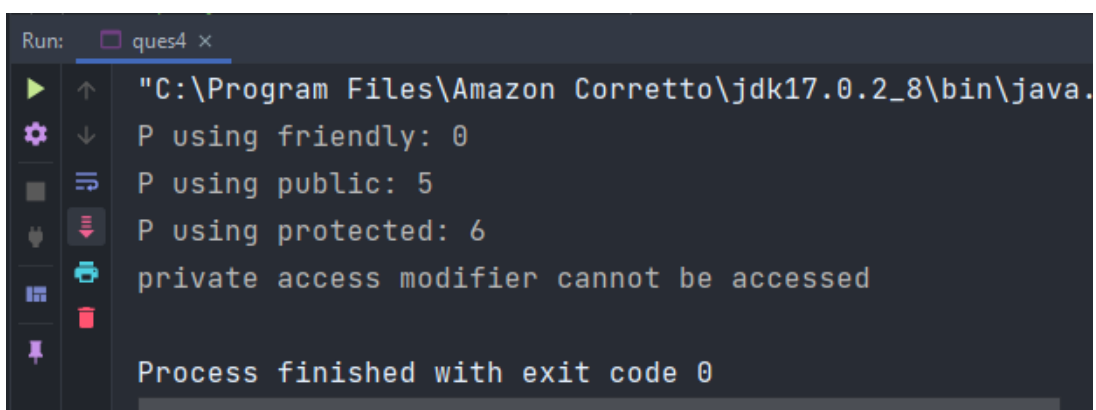
```

```

public class ques4 {
    public static void main (String[] args) {
        A obj1 = new A();
        B obj2 = new B();
        C obj3 = new C();

        System.out.println("P using friendly:" + obj1.PC());
        System.out.println("P using public: " + obj2.PC());
        System.out.println("P using protected:" + obj3.PC());
        System.out.println("Private Access Modifier Cannot be
            Accessed");
    }
}

```



```

Run: ques4 x
"C:\Program Files\Amazon Corretto\jdk17.0.2_8\bin\java.
P using friendly: 0
P using public: 5
P using protected: 6
private access modifier cannot be accessed
Process finished with exit code 0

```


Ques 5

```
public class Abstract Demo
```

```
{
```

```
    public static void main(String[] args)
```

```
    {
```

```
        Shape shape;
```

```
        Circle c1 = new Circle();
```

```
        shape = c1;
```

```
        shape.setValues(5);
```

```
        System.out.println("Area of Circle: " +  
                             shape.getArea());
```

```
        Square sq = new Square();
```

```
        shape = sq;
```

```
        shape.setValues(5);
```

```
        System.out.println("Area of Square: " +  
                             shape.getArea());
```

```
        Triangle tri = new Triangle();
```

```
        shape = tri;
```

```
        shape.setValues(5);
```

```
        System.out.println("Area of Triangle: " + shape.  
                             getArea());
```

```
    }
```

```
public abstract class Shape
```

```
{
```

```
    private double radius;
```

```
    public void setValues(double radius)
```

```
    {
```

```
        this.radius = radius;
```

```
    }
```

```
    public double getRadius()
```

```
    {
```

```
        return radius;
```

```
    }
```

```
    public abstract double getArea();
```

```
}
```

```
public class Square extends Shape
```

```
{
```

```
    public double getArea()
```

```
    {
```

```
        return getRadius() * getRadius();
```

```
    }
```

```
}
```

```
public class Triangle extends Shape {
```

```
    public double getArea()
```

```
{
```

```
    return Math.sqrt(3) * 0.25 * getRadius() getRadius() *  
        getRadius();
```

```
}
```

```
}
```

```
public class Circle extends Shape
```

```
{
```

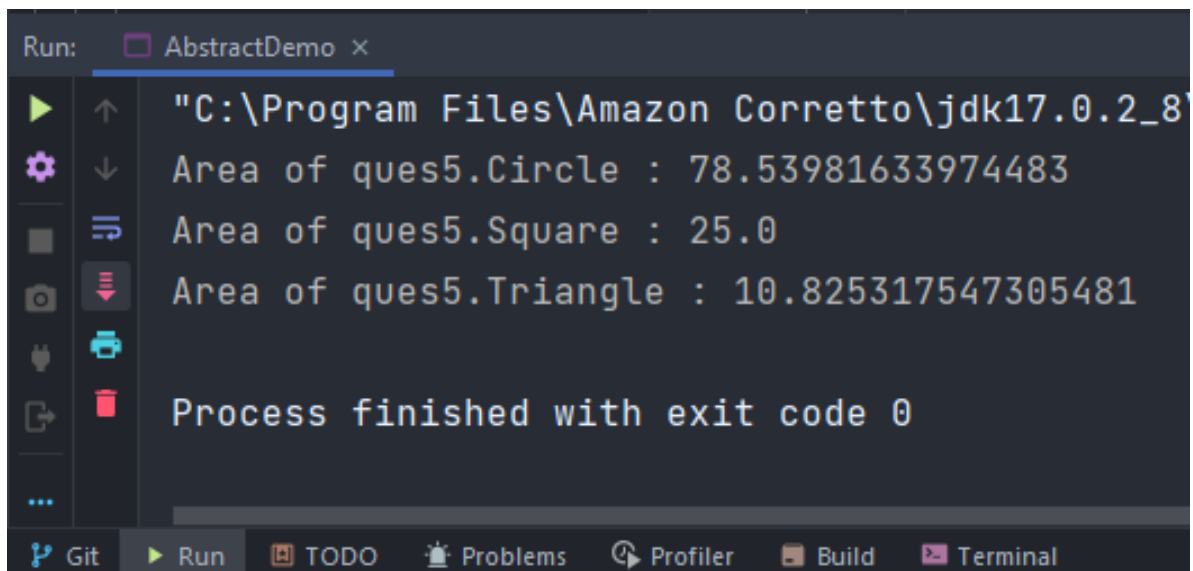
```
    public double getArea()
```

```
{
```

```
    return Math.PI * (getRadius() * getRadius());
```

```
}
```

```
}
```



```
Run: AbstractDemo x  
"C:\Program Files\Amazon Corretto\jdk17.0.2_8"  
Area of ques5.Circle : 78.53981633974483  
Area of ques5.Square : 25.0  
Area of ques5.Triangle : 10.825317547305481  
  
Process finished with exit code 0  
  
Git Run TODO Problems Profiler Build Terminal
```

Ques 6

```
1) public class ques6_1 {  
    public static void main(String[] args) {  
        final int age = 18;  
        age = 15;  
        System.out.println("Age is: " + age);  
    }  
}
```

```
2) public class ques6_2 {  
    class FinalDemo {  
        public final void display() {  
            System.out.println("Final Method")  
        }  
    }  
}
```

```
class Main extends FinalDemo {  
    public final void display() {  
        System.out.println("Overridden Final  
Method");  
    }  
}
```

```

public static void main(String[] args) {
    Main obj = new Main();
    obj.display();
}
}
}

```

```

public class ques 6_3 {

```

```

    final class Final {

```

```

        public void display() {

```

```

            System.out.println("Final Method");

```

```

        }

```

```

    }

```

```

class Main extends FinalClass {

```

```

    public void display() {

```

```

        System.out.println("Final Method
                               Overridden");

```

```

    }

```

```

public static void main(String[] args) {

```

```

    Main obj = new Main();

```

```

    obj.display();

```

```

}

```

```

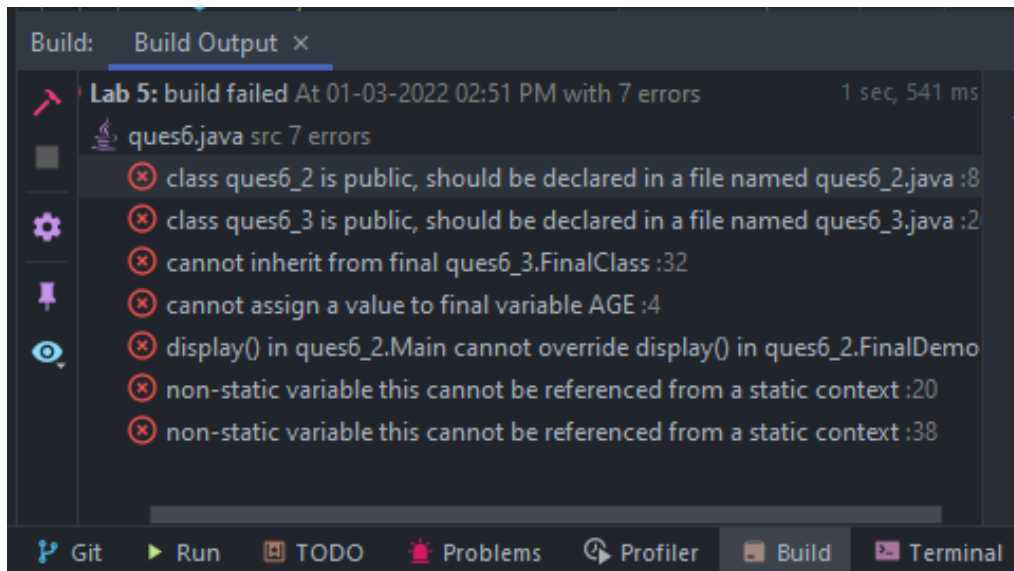
}

```

```

}

```

Ques 7

```
public class ques 7 {  
    public static void main(String[] args) {  
  
        Object object_obj1 = new Object();  
        System.out.println(object_obj1.toString());  
  
        Object object_obj2 = new Object();  
        System.out.println(object_obj2.toString());  
  
        Object object_obj3 = new Object();  
        System.out.println(object_obj3.toString());  
  
        Object object_obj4 = new Object();  
        System.out.println(object_obj4.toString());  
  
    }  
}
```

```
Run: ques7 x
"C:\Program Files\Amazon Corretto\jdk17.0
java.lang.Object@e9e54c2
java.lang.Object@65ab7765
java.lang.Object@1b28cdfa
java.lang.Object@eed1f14

Process finished with exit code 0
```

Ques 8

```
abstract class student {
```

```
    double RegNo = 1234;
```

```
    double RollNo = 523;
```

```
    abstract void course();
```

```
}
```

```
class kiitian extends student {
```

```
    void course() {
```

```
        System.out.println("RollNo:" + RollNo);
```

```
        System.out.println("RegNo:" + RegNo);
```

```
    }
```

```
}
```

```
public class ques8 {
```

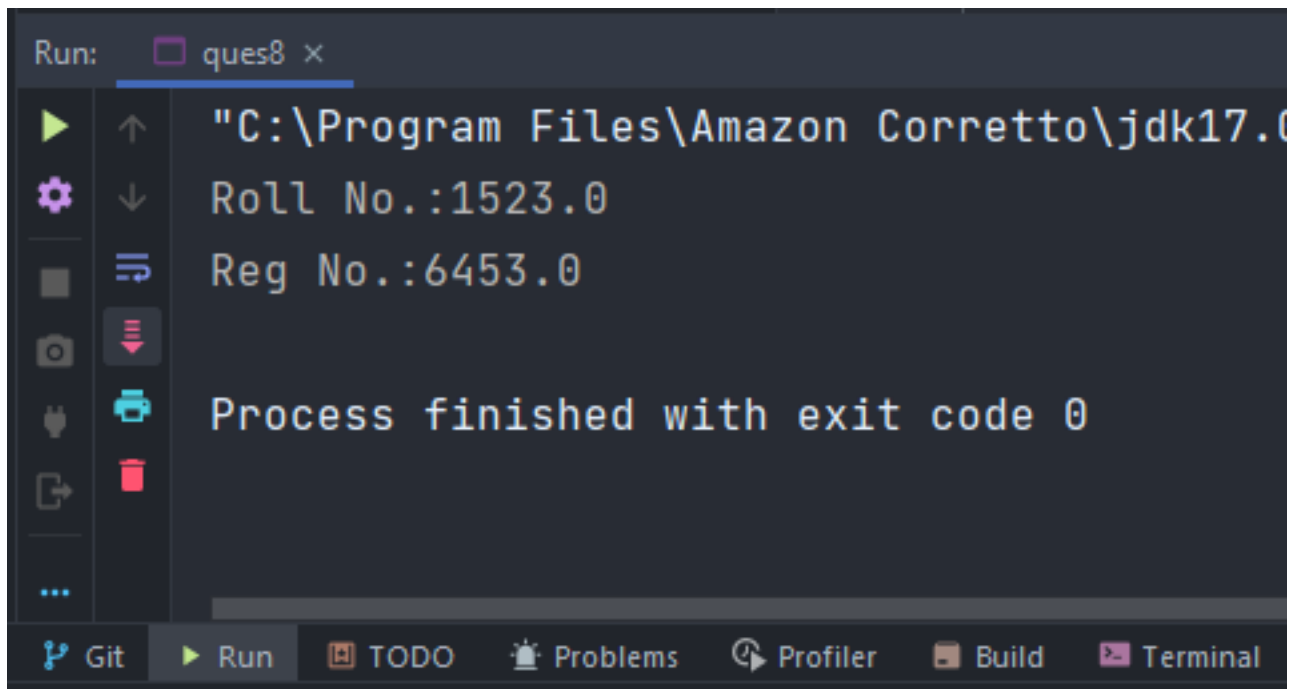
```
    public static void main(String[] args) {
```

```
        student stu = new kiitian();
```


stu.course();

}

}



The screenshot shows the 'Run' console of an IDE. The title bar indicates the active run configuration is 'ques8'. The console output displays the program's execution path and data: the Java runtime path is 'C:\Program Files\Amazon Corretto\jdk17.0', followed by 'Roll No.:1523.0' and 'Reg No.:6453.0'. The final line of output is 'Process finished with exit code 0'. On the left side of the console, there is a vertical toolbar with icons for running, stepping through, and other debugging actions. At the bottom of the IDE window, a tabbed interface shows 'Run' as the active tab, with other tabs like 'Git', 'TODO', 'Problems', 'Profiler', 'Build', and 'Terminal' visible.

```
Run: ques8 x
"C:\Program Files\Amazon Corretto\jdk17.0
Roll No.:1523.0
Reg No.:6453.0
Process finished with exit code 0
```

Git Run TODO Problems Profiler Build Terminal