Department of Computing
Faculty of Computing and Information Technology (FCIT),Indus University, Karachi

NAME OF STUDENT: Muhammad Owais Qadri **ID No: 519-2020**

FINAL EXAMINATION

FALL 2020

Department / Faculty	Program	Semester	Course Title	Instructor	Date of Exam	Time Allowed	Total Marks
(Computing) / FCIT	BS Computer science	2nd	Object Oriented Programming (3+0)	Aadrish Pirzado	10-02-2021	24 Hours 12:00 ~AM 2:00 PM	40

Instructions:

1. This paper contains **2Questions**. Attempt all questions.
2. Marks of each question are mentioned at the end of each question.
3. Use of Calculator is allowed.
4. Mobile Phones must be powered off and kept away during the exam.
5. Cheating of any type will disqualify the candidate.
6. Admit card is compulsory to appear in exam.
7. Student shall not be allowed leave exam hall before the specified time.
8. Student shall not be allowed to enter in exam hall after the specified time.
9. Student must comply with given instructions by the invigilator or examiner.
10. Any arguments/misbehavior with invigilator shall be resulted to disciplinary action.
11. Do not detach the sheets. (Paper will be cancelled, if the sheets are detached).
12. Write your answers in ink. Pencil may be used for underlining of diagrams only.
13. The work must be neat & clean. Over-writing, cutting will be considered as mistake.
14. Student must bring his/her own stationary. Borrowing in exam hall is not allowed.
15. Answer Script & Question Paper must be returned back to invigilator before leaving exam hall.

This table is for official use; do not write anything on it.

CLOs	CLO_1			CLO_2		
Question Number	1	1		2	2	
Student's Score						
Maximum Score	10	10		10	10	40

This paper has a total of **4 pages** including this title page

Department of Computing

Faculty of Computing and Information Technology (FCIT),Indus University, Karachi

NAME OF STUDENT: Muhammad Owais Qadri **ID No: 519-2020**

FINAL EXAMINATION FALL 2020

Object Oriented Programming (3+0)

Dated: 10-02-2021

Question 1 is related to CLO # 1:]

CLO-1: Understand basic concepts of objects and classes.

Question 1(a): **Define** packages and its usage by making a program. **10**

Question (b): **Discuss** access modifier and its types. **10**

[Question 2 is related to CLO # 2:]

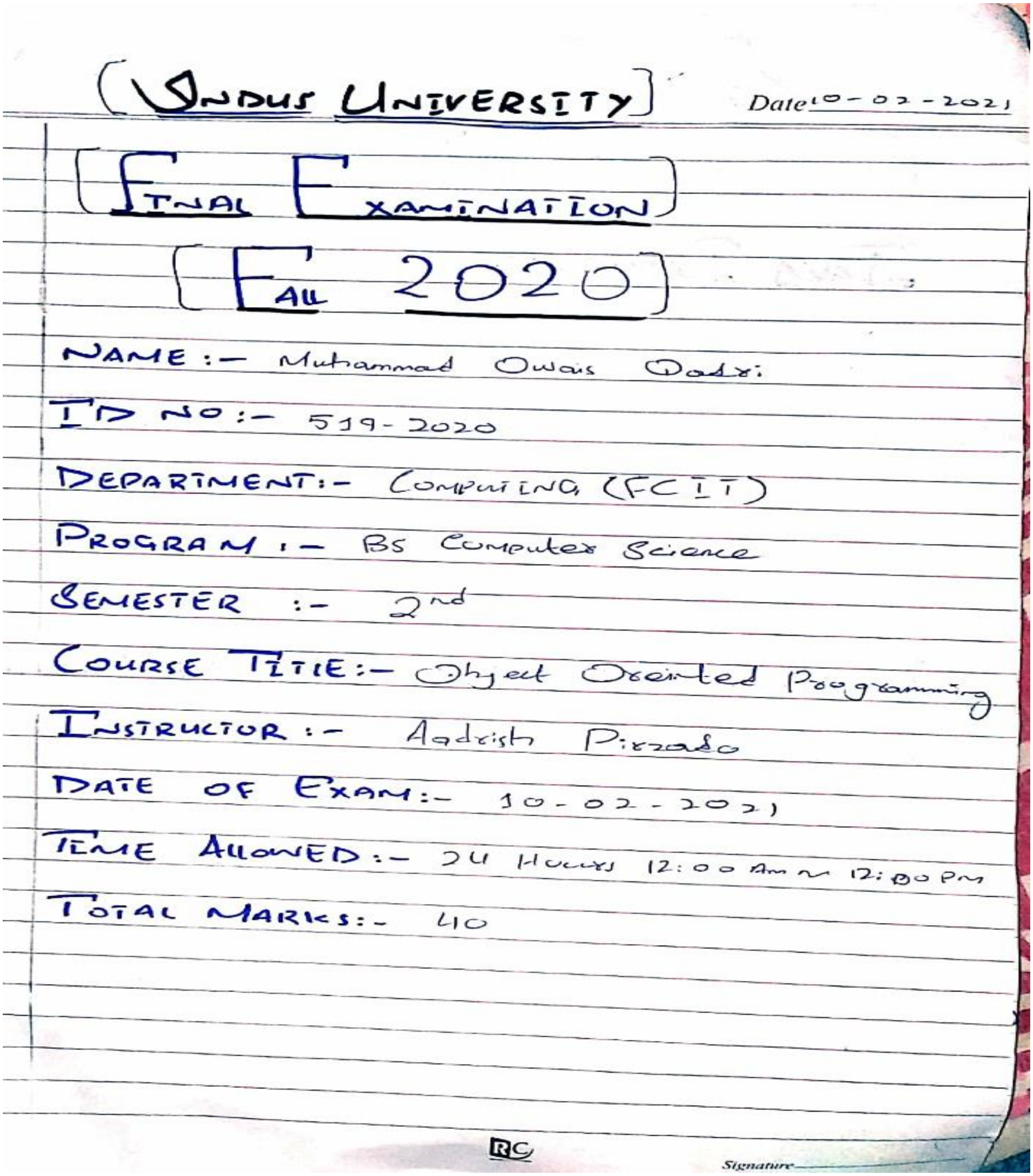
CLO-2: Apply basics object oriented programming concepts.

Question 2(a): Why we use java static keyword **Describe** by making a program. **10**

Question 2(b): Make a school management system by **applying** object Oriented techniques. **10**

Examiner:

NAME OF STUDENT: **Muhammad Owais Qadri** **ID No:** **519-2020**



(Indus University) Date 10-02-2021

(Final Examination)

(Fall 2020)

NAME :- Muhammad Owais Qadri

ID NO :- 519-2020

DEPARTMENT:- Computing (FCIT)

PROGRAM :- BS Computer Science

SEMESTER :- 2nd

COURSE TITLE:- Object Oriented Programming

INSTRUCTOR :- Aadrish Pirzada

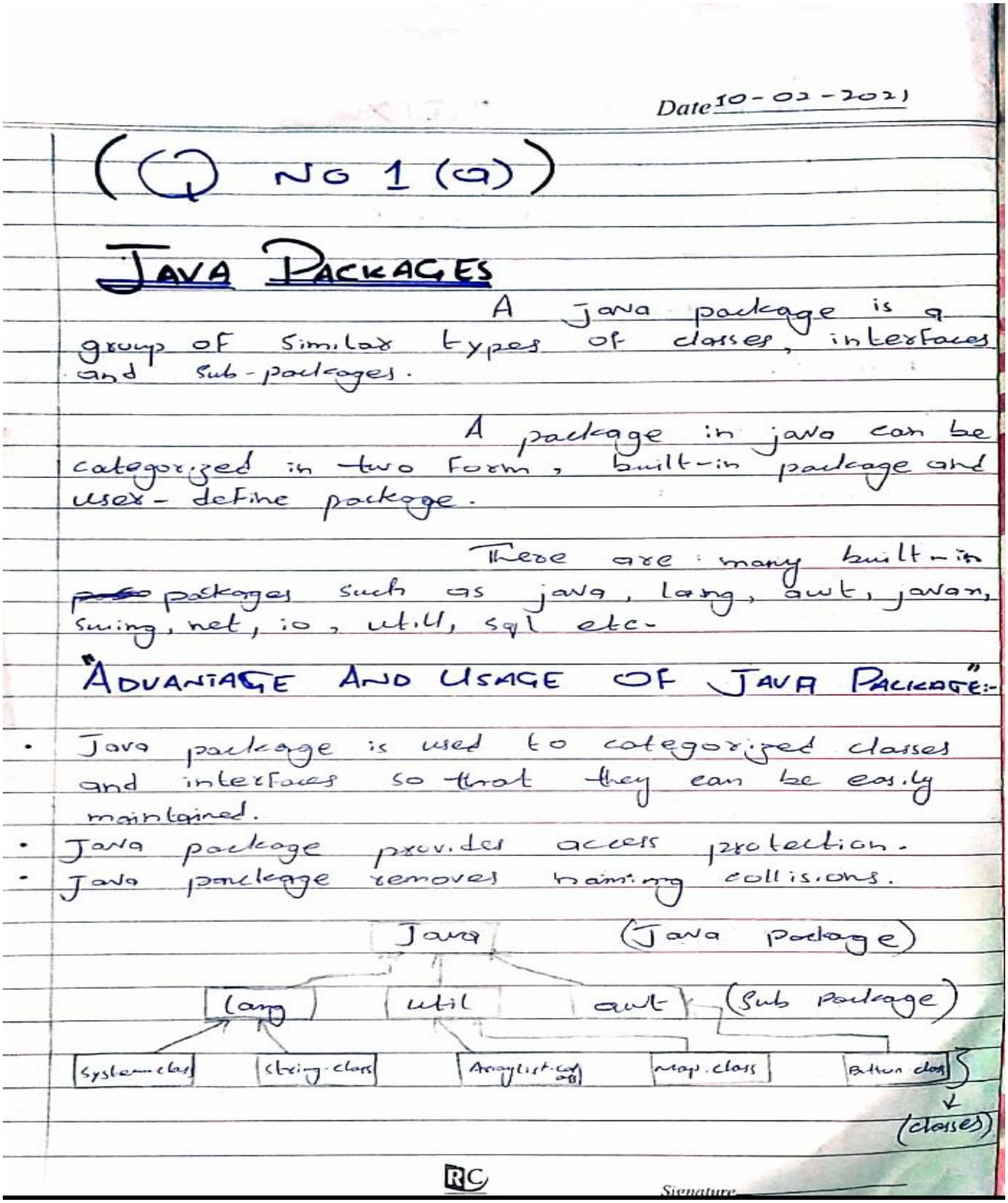
DATE OF EXAM:- 10-02-2021

TIME ALLOWED:- 24 Hours 12:00 Am ~ 12:00 PM

TOTAL MARKS:- 40

Signature

NAME OF STUDENT: Muhammad Owais Qadri **ID No: 519-2020**



Date 10-02-2021

(Q No 1 (a))

Java Packages

A java package is a group of similar types of classes, interfaces and sub-packages.

A package in java can be categorized in two form, built-in package and user-define package.

There are many built-in ~~pack~~ packages such as java, lang, awt, javax, swing, net, io, util, sql etc.

"Advantage And Usage Of Java Package":-

- Java package is used to categorized classes and interfaces so that they can be easily maintained.
- Java package provides access protection.
- Java package removes naming collisions.

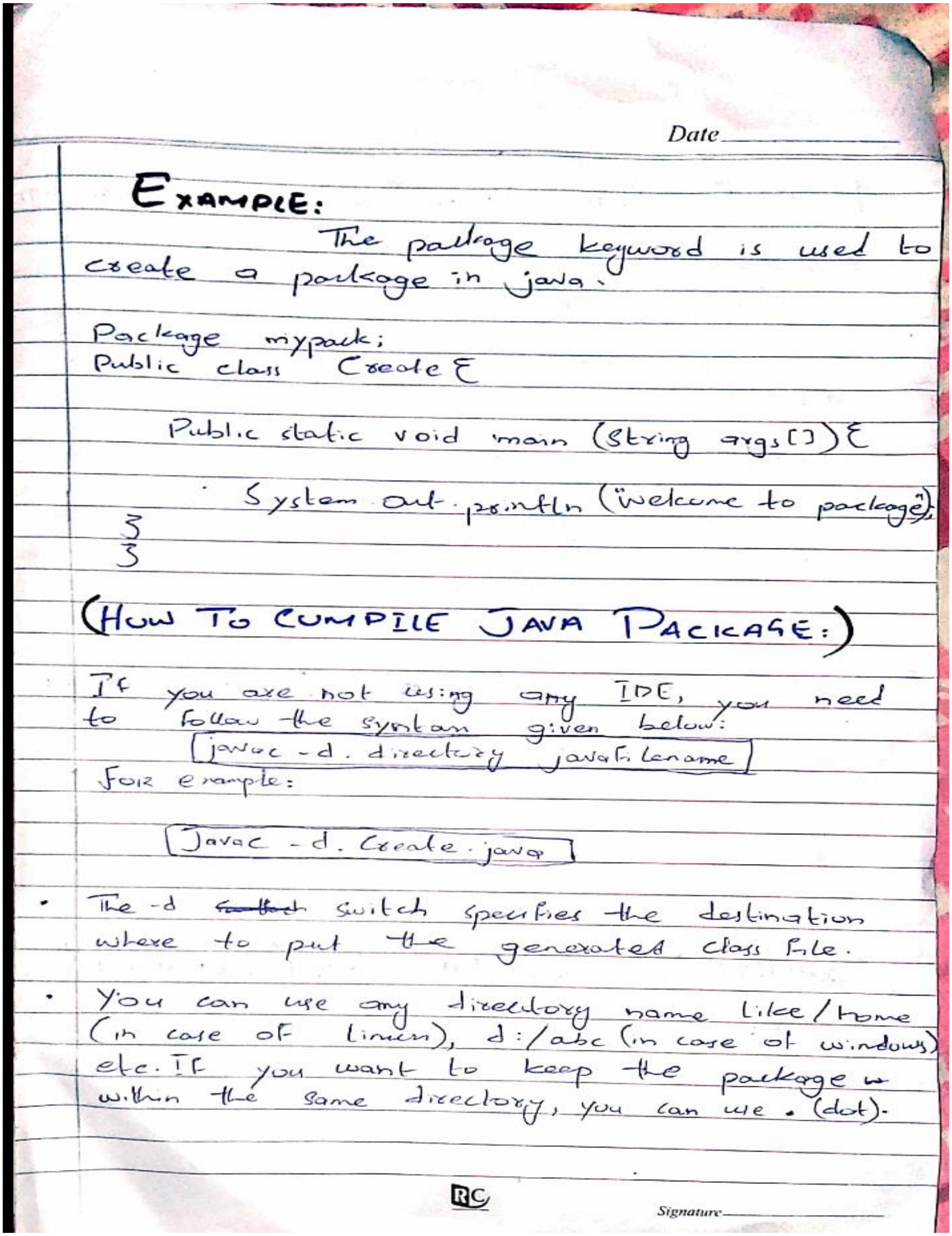
Java (Java package)

lang util awt (Sub Package)

System.class String.class Arraylist.class Map.class Button class

(classes)

Signature



Date

EXAMPLE:

The package keyword is used to create a package in java.

```
Package mypack;
Public class Create {

    Public static void main (String args[]){

        System out.println ("welcome to package");
}
}
```

(HOW TO COMPILE JAVA PACKAGE:)

If you are not using any IDE, you need to follow the syntax given below:

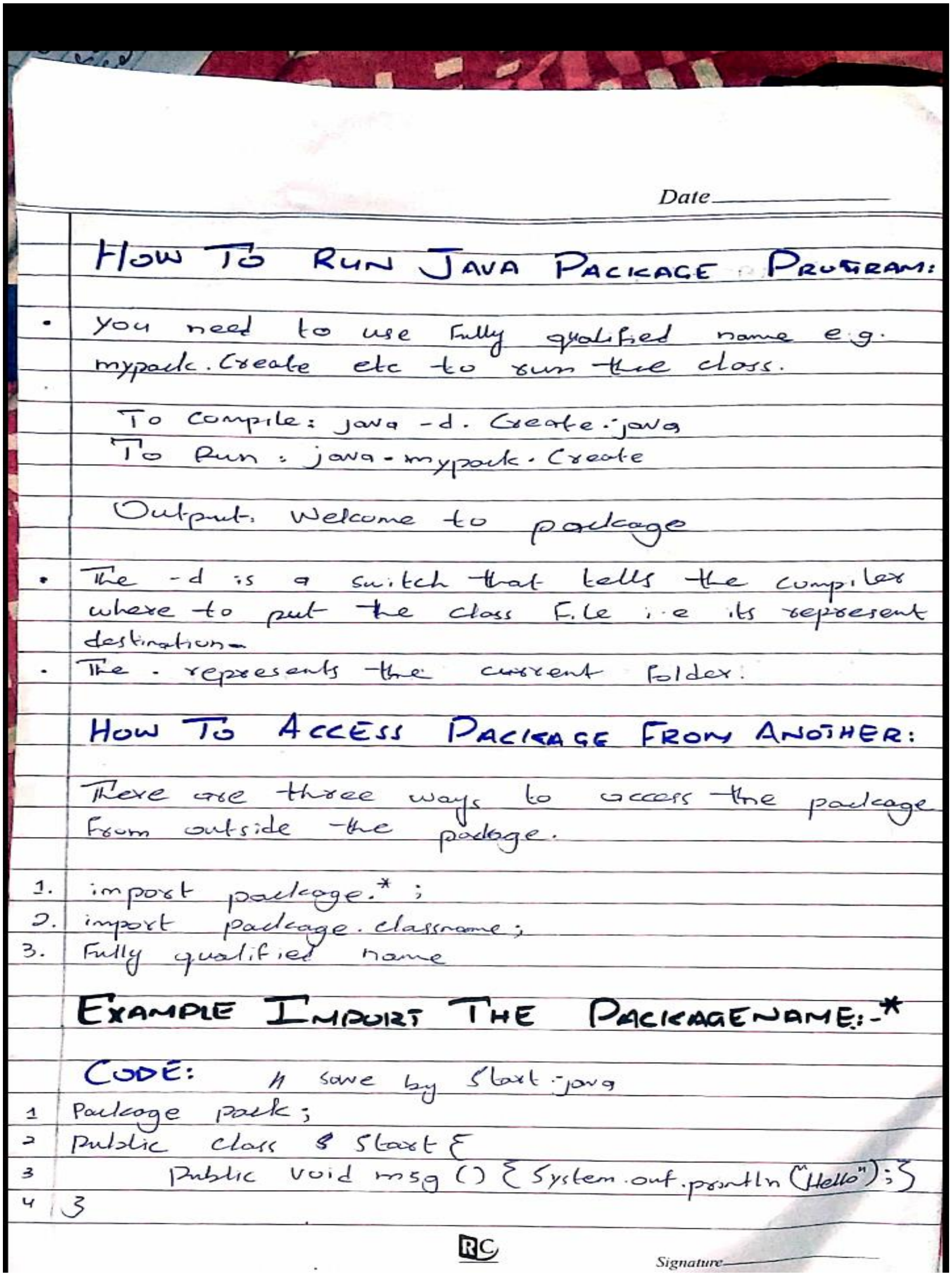
[javac -d . directory javafilename]

For example:

[Javac -d . Create.java]

- The -d switch specifies the destination where to put the generated class file.
- You can use any directory name like /home (in case of linux), d:/abc (in case of windows) etc. If you want to keep the package within the same directory, you can use . (dot).

Signature



Date__________

HOW TO RUN JAVA PACKAGE PROGRAM:

- you need to use Fully qualified name e.g. mypack.Create etc to run the class.

To compile: java -d. Create.java
To Run: java - mypack.Create

Output: Welcome to package

- The -d is a switch that tells the compiler where to put the class File i.e its represent destination.
- The . represents the current folder.

HOW TO ACCESS PACKAGE FROM ANOTHER:

There are three ways to access the package from outside the package.

1. import package.* ;
2. import package.classname;
3. Fully qualified name

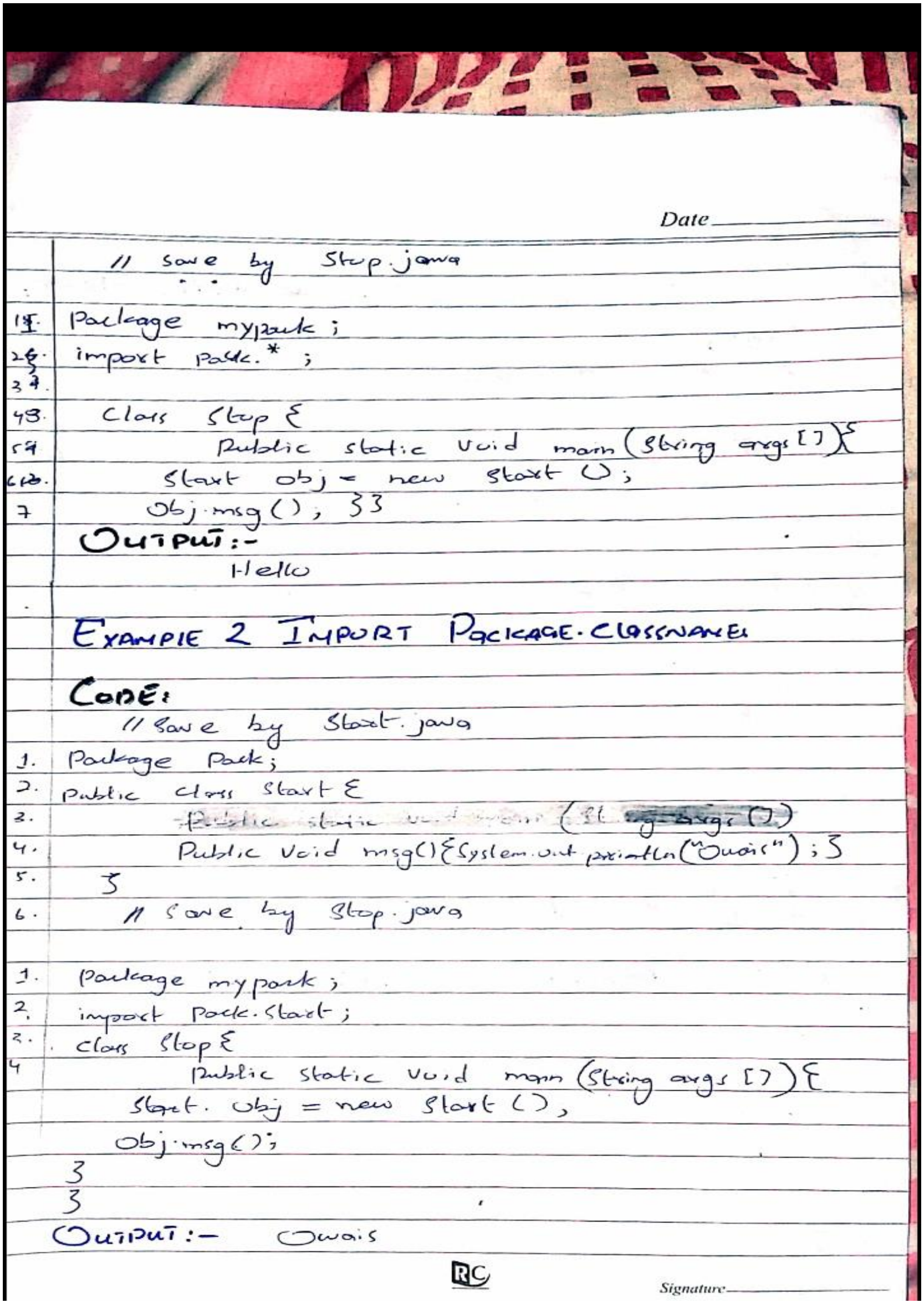
EXAMPLE IMPORT THE PACKAGENAME:-*

CODE: // save by Start.java

```
Package pack;
Public class $ Start {
        Public void msg() { System.out.println("Hello"); }
}
```

Signature__________

NAME OF STUDENT: **Muhammad Owais Qadri** **ID No:** **519-2020**



Date

```
// save by Stop.java

Package mypack ;
import pack.* ;

Class Stop {
        Public static void main(String args[]){
    Start obj = new start ();
    Obj.msg(); }}
```

OUTPUT:-

```
Hello
```

EXAMPLE 2 IMPORT PACKAGE.CLASSNAME

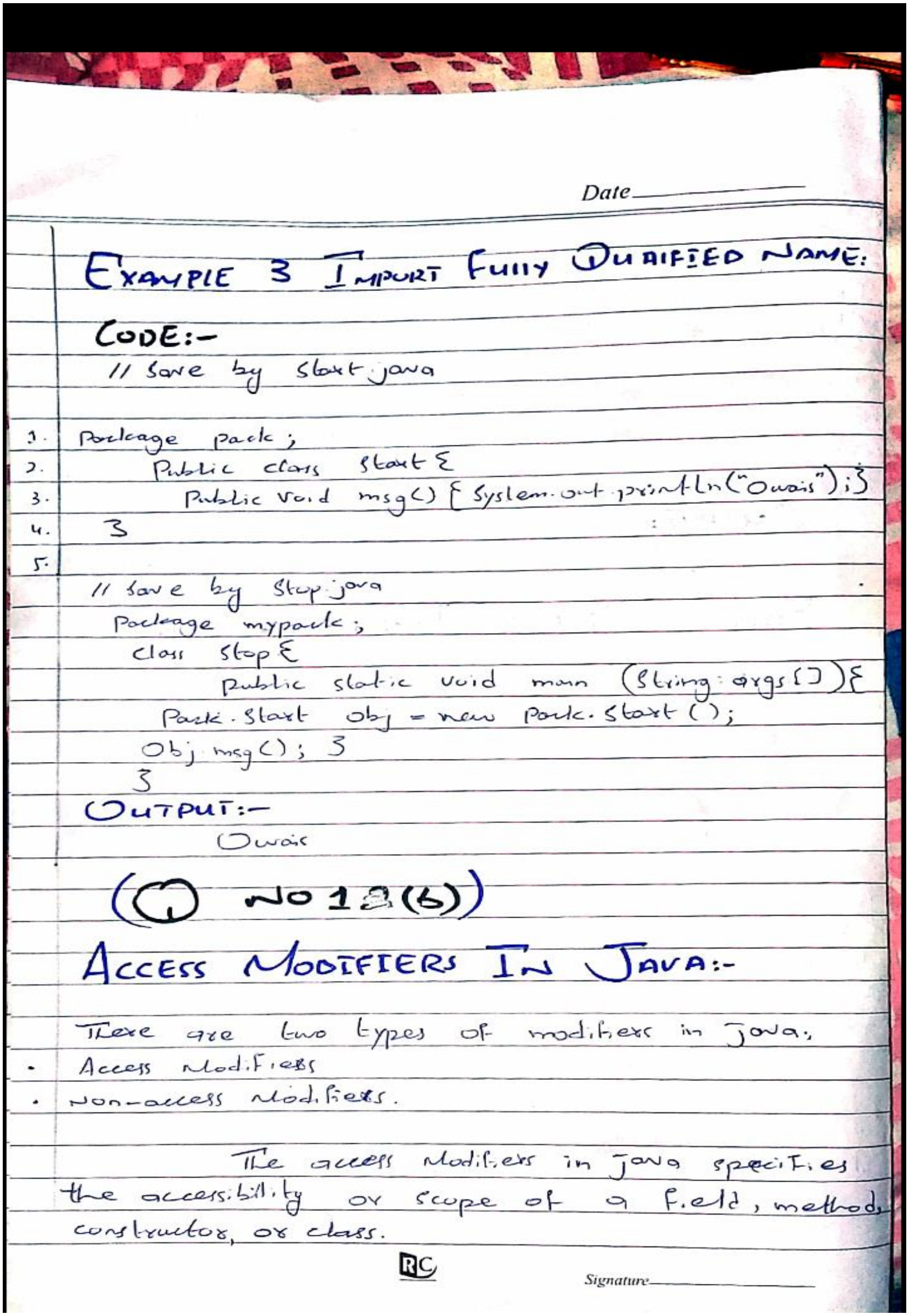
CODE:

```
    // Save by Start.java
Package Pack;
public class Start {
        ~~Public static void main (String args [])~~
        Public Void msg(){System.out.println("Owais");}
  }
    // Save by Stop.java

Package mypack ;
import Pack.Start ;
class Stop {
        Public static void main (String args []){
    Start. obj = new Start (),
    Obj.msg();
  }
  }
```

OUTPUT:- Owais

RC

Signature



Date

EXAMPLE 3 IMPORT FUlly QUAlIFIED NAME:

CODE:-

```
// Save by Start.java

Package pack;
    Public class Start {
      Public Void msg() { System.out.println("Owais");}
 }

// save by Stop.java
 Package mypack;
  class Stop {
          public static void main (String: args[] ){
     Pack.Start obj = new Pack.Start ();
   Obj.msg(); }
   }
```

OUTPUT:-

```
        Owais
```

(Q NO 1[illegible](b))

ACCESS MODIFIERS IN JAVA:-

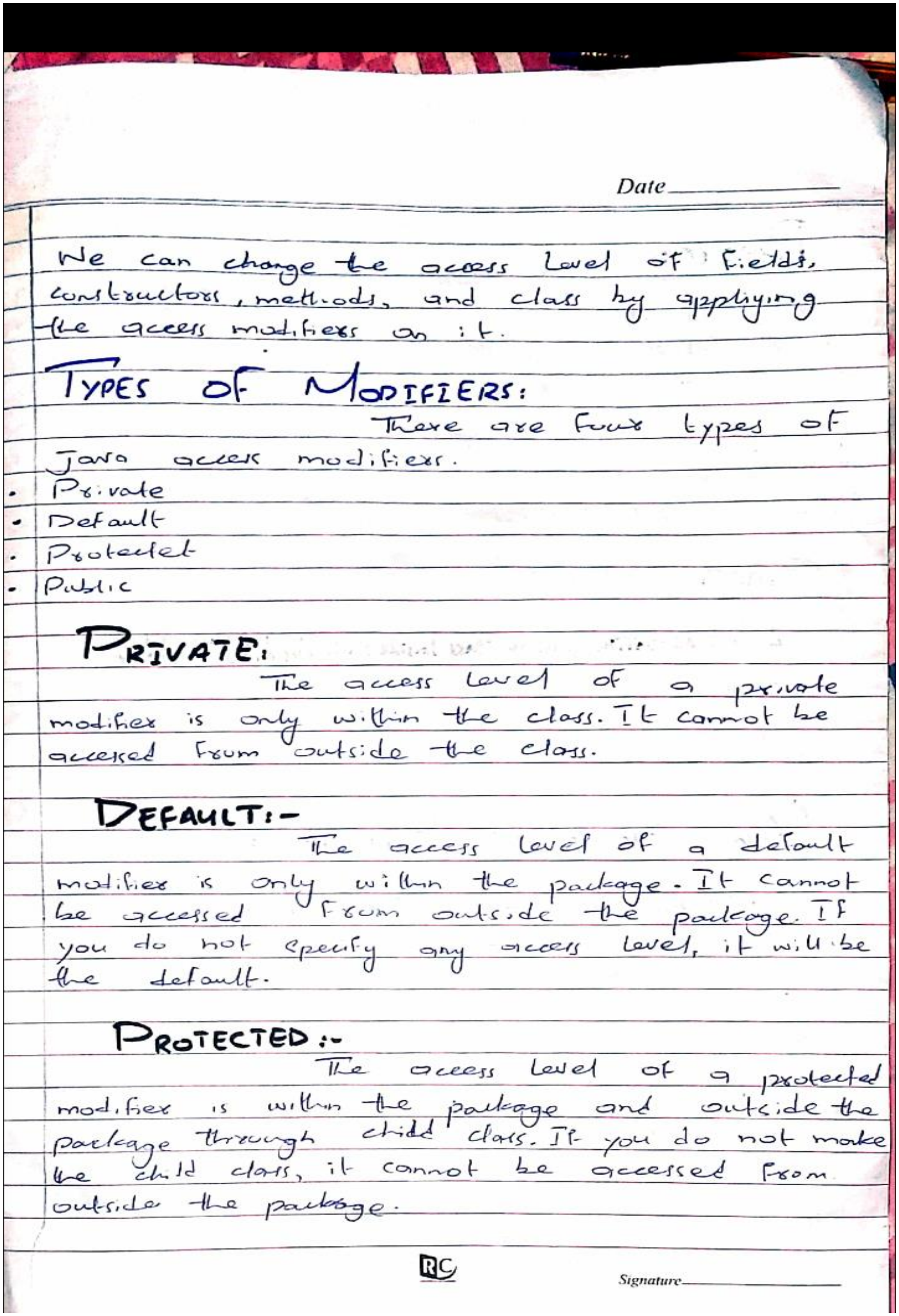
There are two types of modifiers in Java;

- Access Modifiers
- Non-access Modifiers.

The access Modifiers in Java specifies the accessibility or scope of a field, method, constructor, or class.

RC

Signature



Date

We can change the access level of fields, constructors, methods, and class by applying the access modifiers on it.

TYPES OF MODIFIERS:

There are four types of Java access modifiers.

- Private
- Default
- Protectet
- Public

PRIVATE:

The access level of a private modifier is only within the class. It cannot be accessed from outside the class.

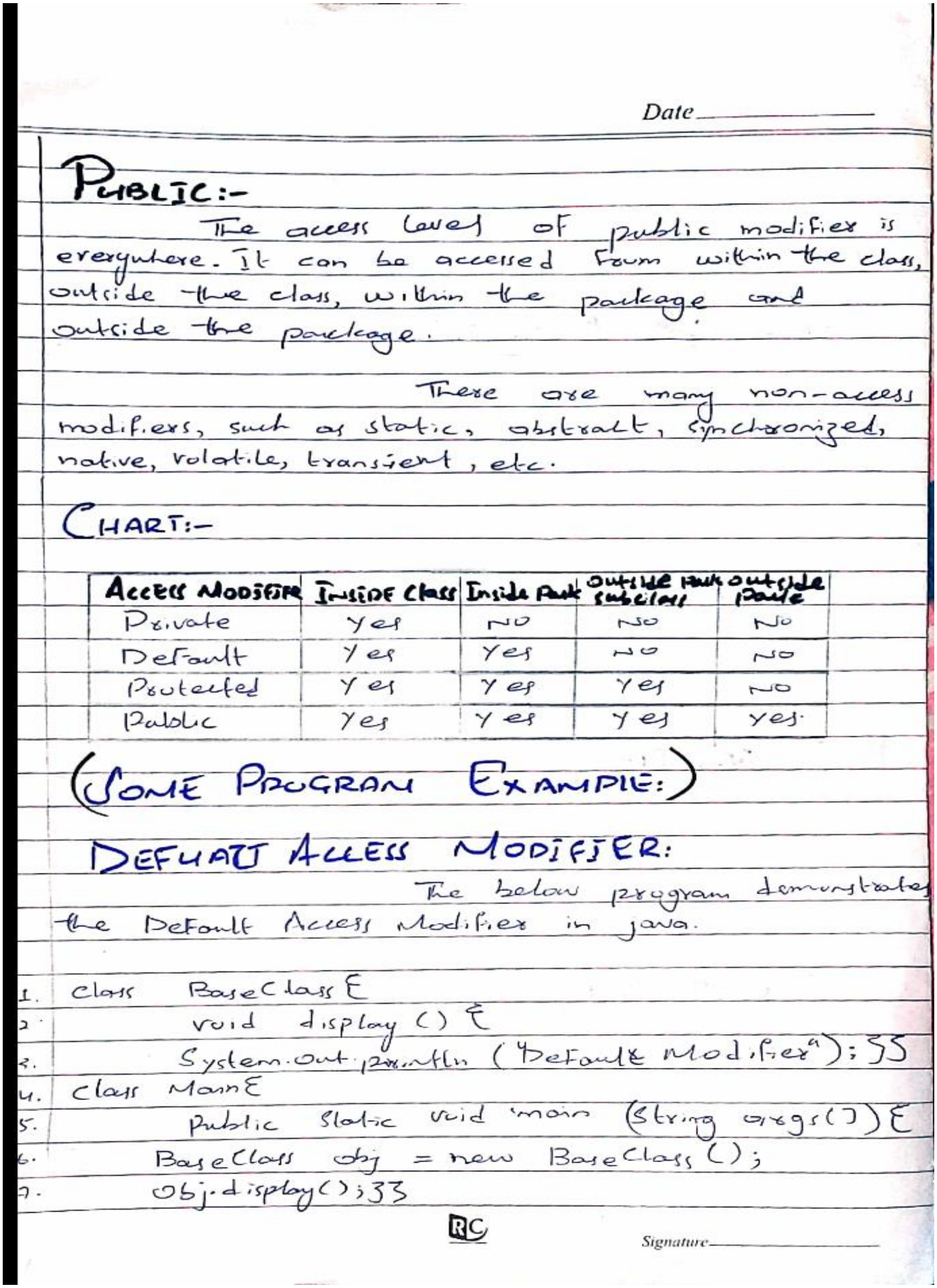
DEFAULT:-

The access level of a default modifier is only within the package. It cannot be accessed from outside the package. If you do not specify any access level, it will be the default.

PROTECTED:-

The access level of a protected modifier is within the package and outside the package through child class. If you do not make the child class, it cannot be accessed from outside the package.

Signature



Date

PUBLIC:-

The access level of public modifier is everywhere. It can be accessed from within the class, outside the class, within the package and outside the package.

There are many non-access modifiers, such as static, abstract, synchronized, native, volatile, transient, etc.

CHART:-

ACCESS MODIFIER	INSIDE CLASS	Inside Pack	Outside Pack subclass	Outside Pack
Private	Yes	NO	NO	NO
Default	Yes	Yes	NO	NO
Protected	Yes	Yes	Yes	NO
Public	Yes	Yes	Yes	Yes.

(SOME PROGRAM EXAMPLE:)

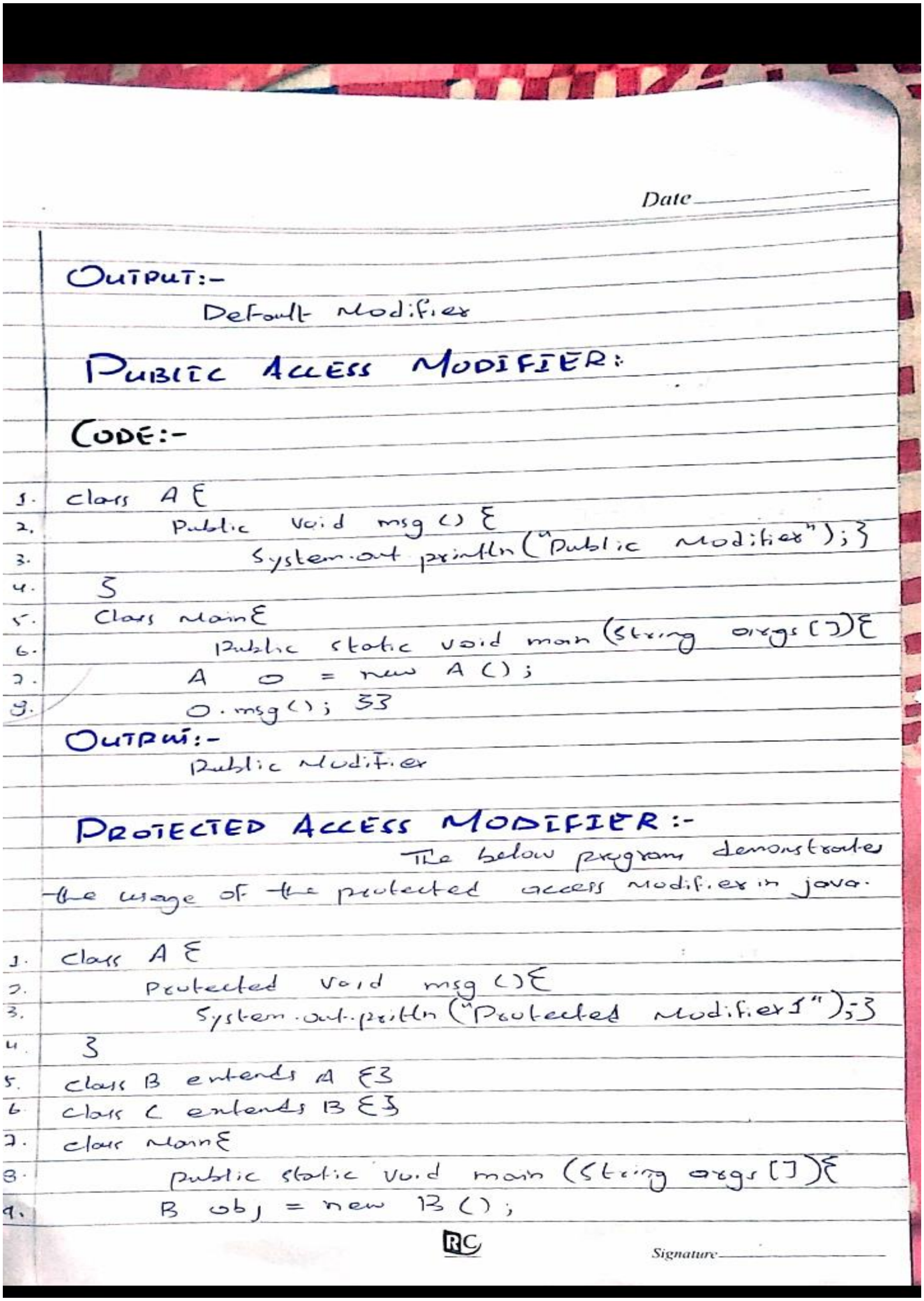
DEFUALT ACCESS MODIFIER:

The below program demonstrates the Default Access Modifier in java.

```
class    BaseClass {
             void    display () {
            System.out.println ("Default Modifier"); }}
class  Main{
             Public    Static    void  main   (String  args[]){
         BaseClass   obj   = new   BaseClass ();
         obj.display();}}
```

RC

Signature



Date

OUTPUT:-

Default Modifier

PUBLIC ACCESS MODIFIER:

CODE:-

```
class A {
        Public void msg () {
             System.out println("Public Modifier");}
  }
  Class Main{
        Public static void main (String args[]){
        A o = new A ();
        o.msg(); }}
```

OUTPUT:-

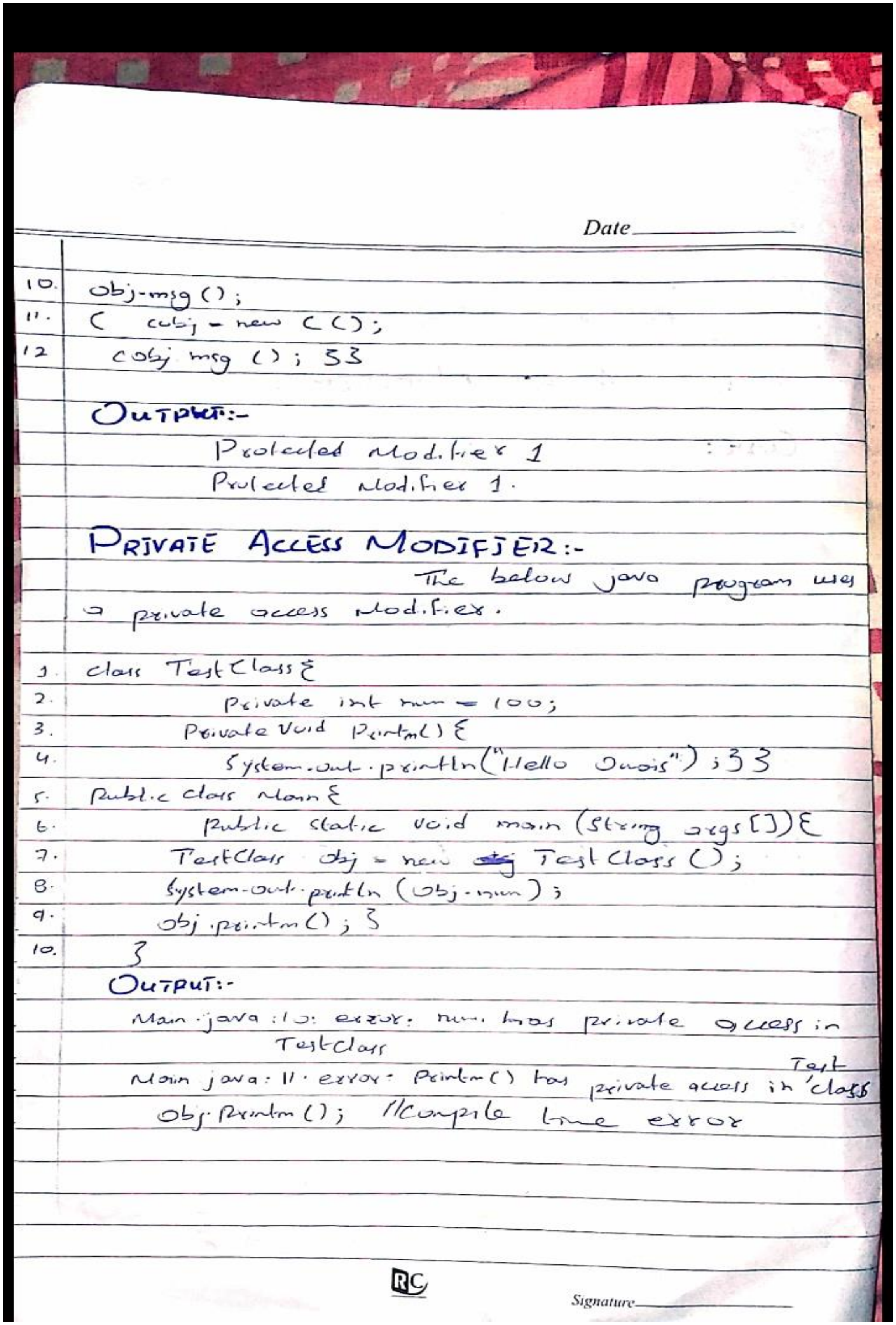
```
        Public Modifier
```

PROTECTED ACCESS MODIFIER:-

The below program demonstrates the usage of the protected access modifier in java.

```
class A {
      Protected void msg (){
          System.out.println("Protected Modifier1");}
 }
class B extends A {}
class C extends B {}
class Main{
      public static void main (String args[]){
      B obj = new B ();
```

Signature



Date

```
10. obj-msg ();
11. C   cobj - new C();
12.   cobj msg (); }}
```

OUTPUT:-

```
Protected Modifier 1
Protected Modifier 1.
```

PRIVATE ACCESS MODIFIER:-

The below java program uses a private access modifier.

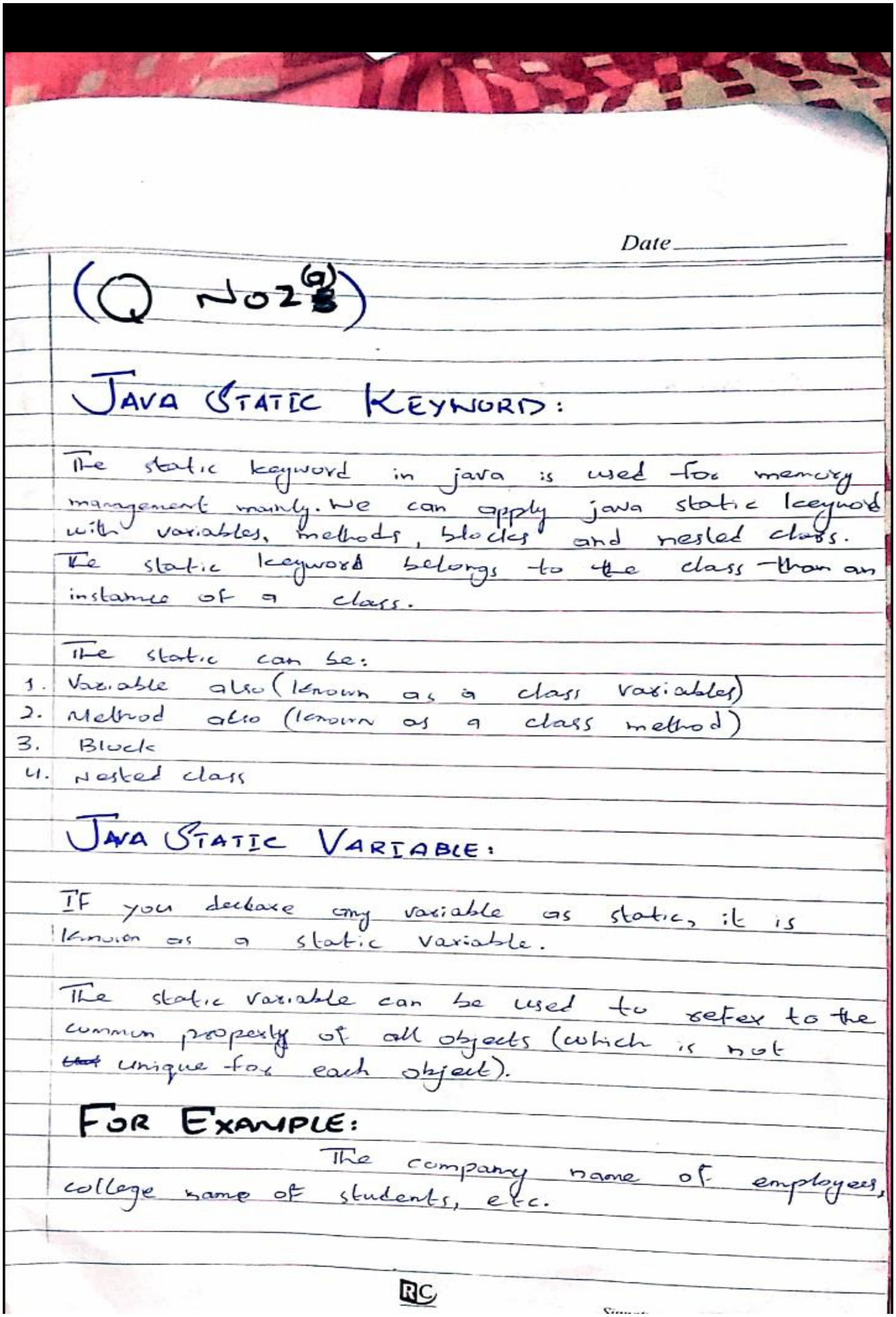
```
1.  class TestClass {
2.          private int num = 100;
3.        Private Void Printm() {
4.          System.out.println("Hello Owais"); }}
5.  Public class Main {
6.        Public static void main (String args[]) {
7.      TestClass obj = new ~~obj~~ TestClass ();
8.      System.out.println (obj.num);
9.      obj.printm(); }
10.   }
```

OUTPUT:-

```
Main.java:10: error: num has private access in
        TestClass
Main java: 11. error: Printm() has private access in Test class
  obj.Printm(); //Compile time error
```

Signature

NAME OF STUDENT: Muhammad Owais Qadri **ID No: 519-2020**



Date

(Q No2(a))

JAVA STATIC KEYWORD:

The static keyword in java is used for memory management mainly. We can apply java static keyword with variables, methods, blocks and nested class. The static keyword belongs to the class than an instance of a class.

The static can be:

1. Variable also (known as a class variables)
2. Method also (known as a class method)
3. Block
4. Nested class

JAVA STATIC VARIABLE:

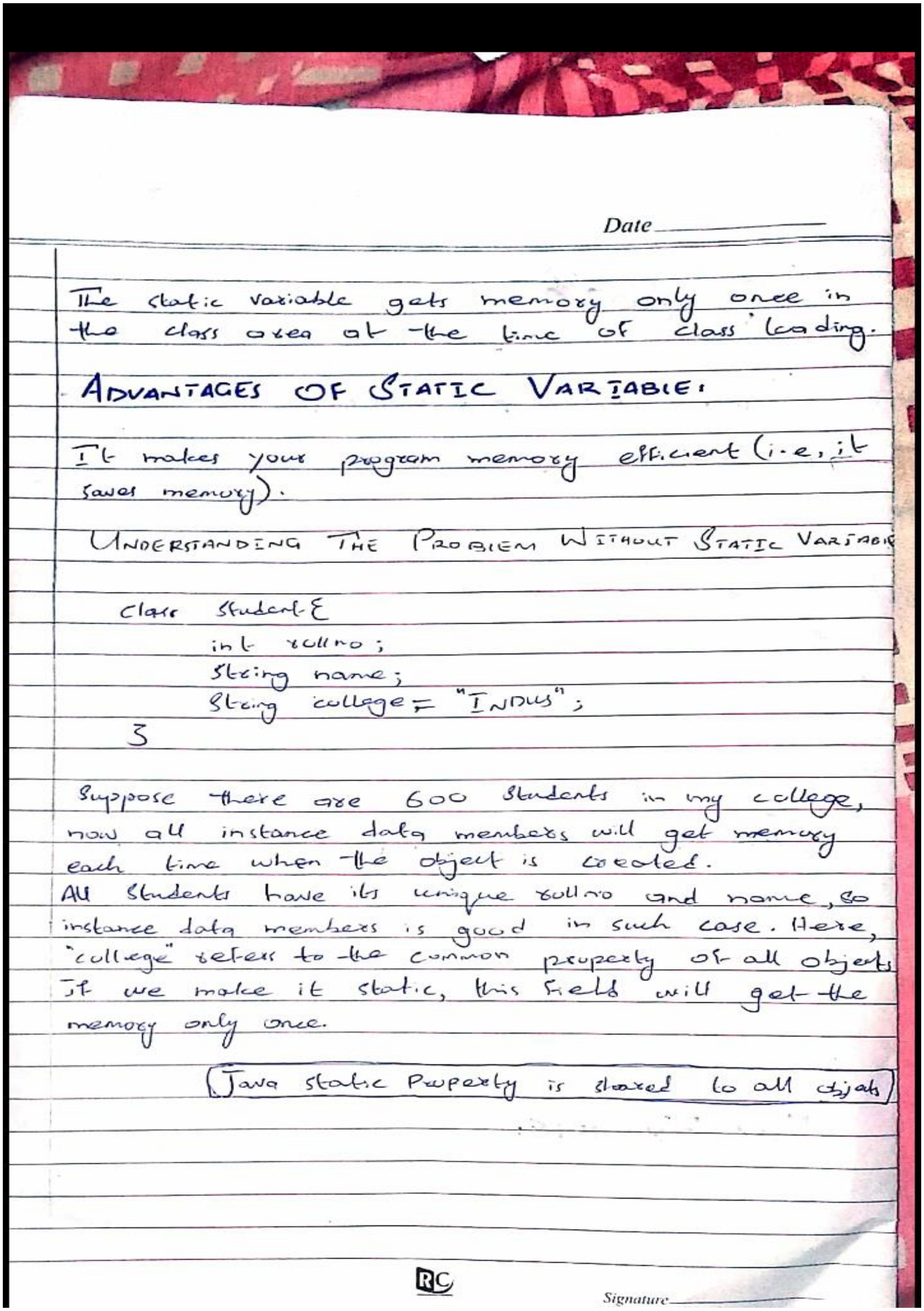
If you declare any variable as static, it is known as a static variable.

The static variable can be used to refer to the common property of all objects (which is not ~~that~~ unique for each object).

FOR EXAMPLE:

The company name of employees, college name of students, etc.

NAME OF STUDENT: Muhammad Owais Qadri **ID No: 519-2020**



Date

The static variable gets memory only once in the class area at the time of class loading.

ADVANTAGES OF STATIC VARIABIE:

It makes your program memory efficient (i.e, it saves memory).

UNDERSTANDING THE PROBIEM WITHOUT STATIC VARIABIE

```
class Student{
        int rollno;
        String name;
        String college= "INDUS";
    }
```

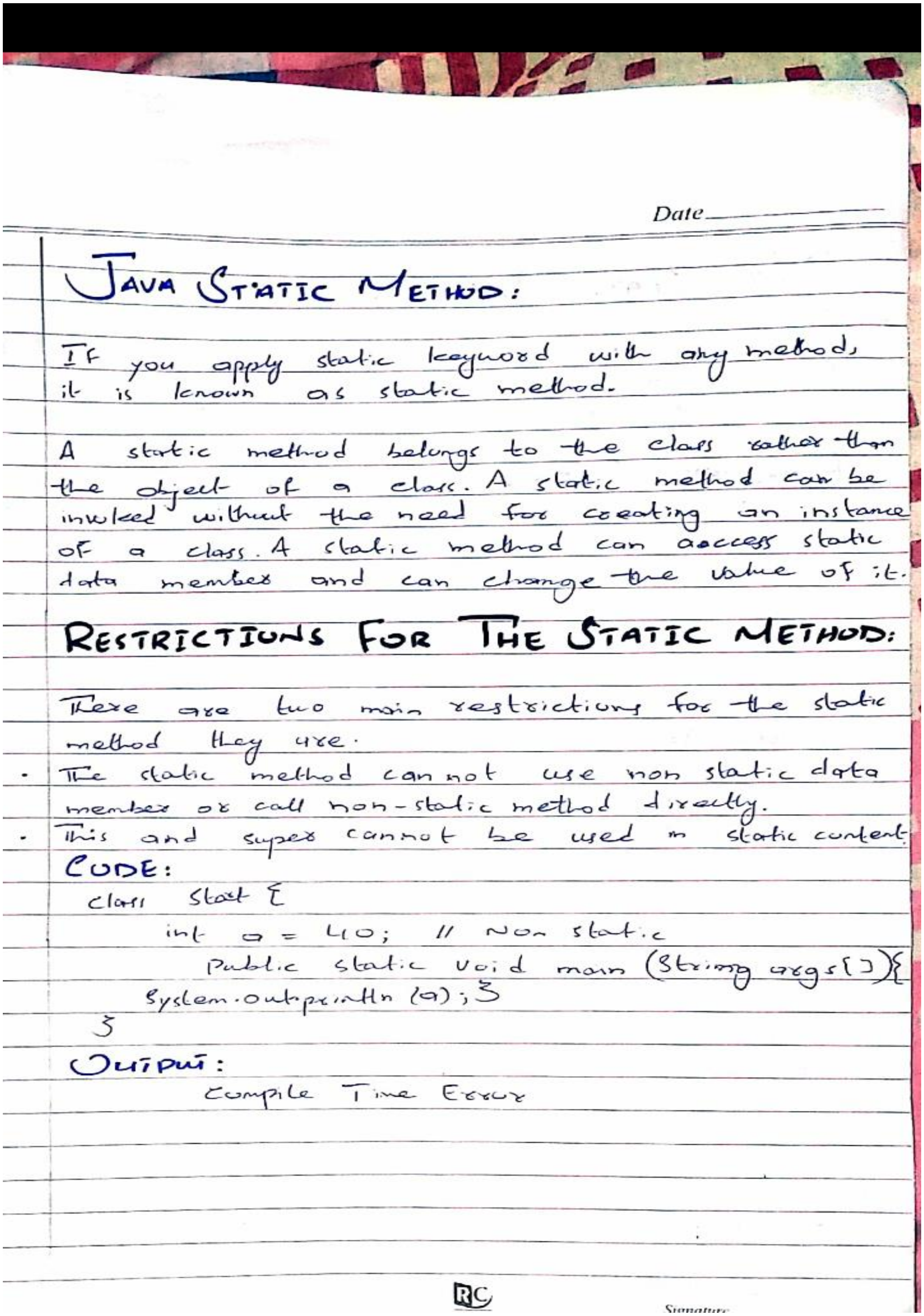
Suppose there are 600 Students in my college, now all instance data members will get memory each time when the object is created. All Students have its unique roll no and name, so instance data members is good in such case. Here, "college" refers to the common property of all objects. If we make it static, this field will get the memory only once.

(Java static Property is shared to all objects)

Signature

NAME OF STUDENT: Muhammad Owais Qadri **ID No: 519-2020**

[redacted]



Date____

Java Static Method:

If you apply static keyword with any method, it is known as static method.

A static method belongs to the class rather than the object of a class. A static method can be inwked without the need for creating an instance of a class. A static method can access static data member and can change the value of it.

Restrictions for the Static Method:

There are two main restrictions for the static method they are.

- The static method can not use non static data member or call non-static method directly.
- This and super cannot be used in static context

Code:

```
class Stat {
        int a = 40;  // Non static
            Public static void main (String args[]){
    System.out.println (a); }
 }
```

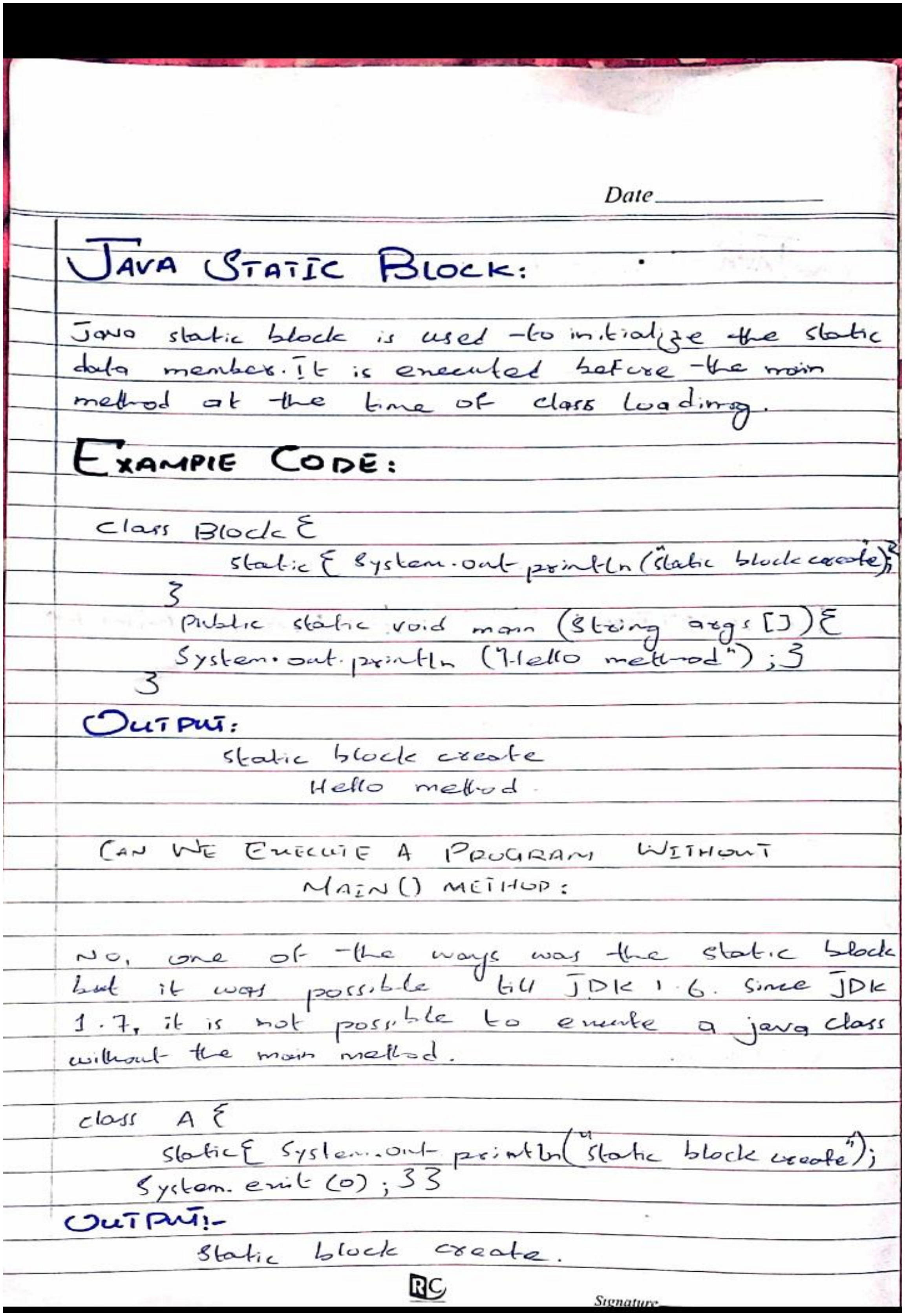
Output:

```
Compile Time Error
```

RC

Signature

NAME OF STUDENT: **Muhammad Owais Qadri** **ID No:** **519-2020**



Date

Java Static Block:

Java static block is used to initialize the static data member. It is executed before the main method at the time of class loading.

Example Code:

```
class Block {
        static { System.out.println("static block create");}
   }
    Public static void main (String args[]) {
    System.out.println ("Hello method") ; }
  }
```

Output:

```
        static block create
              Hello method.
```

Can We Execute A Program Without Main() Method:

No, one of the ways was the static block but it was possible till JDK 1.6. Since JDK 1.7, it is not possible to execute a java class without the main method.

```
class  A {
      static{ System.out.println("static block create");
    System.exit (0) ; }}
```

Output:-

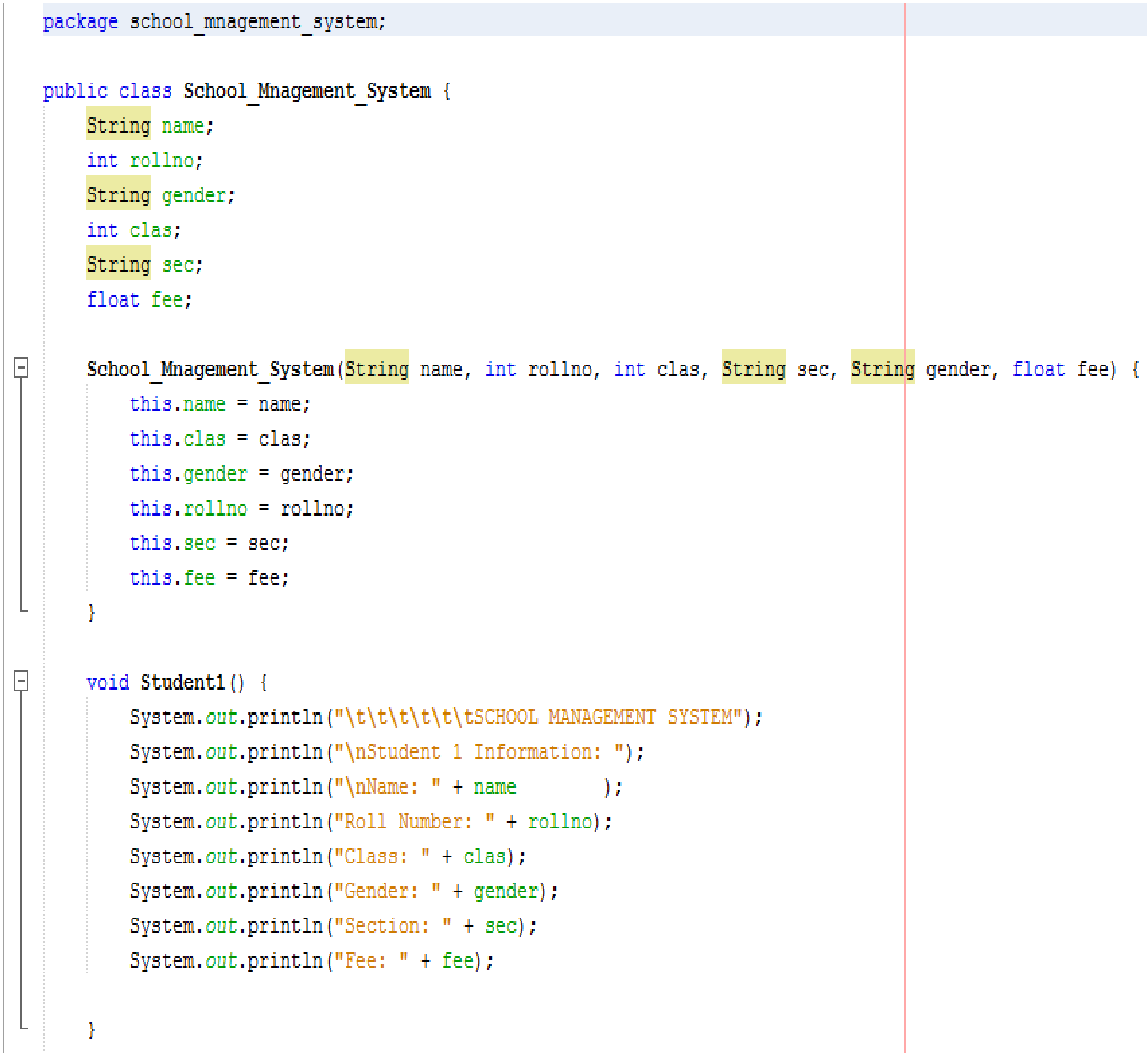
```
      static block create.
```

Signature

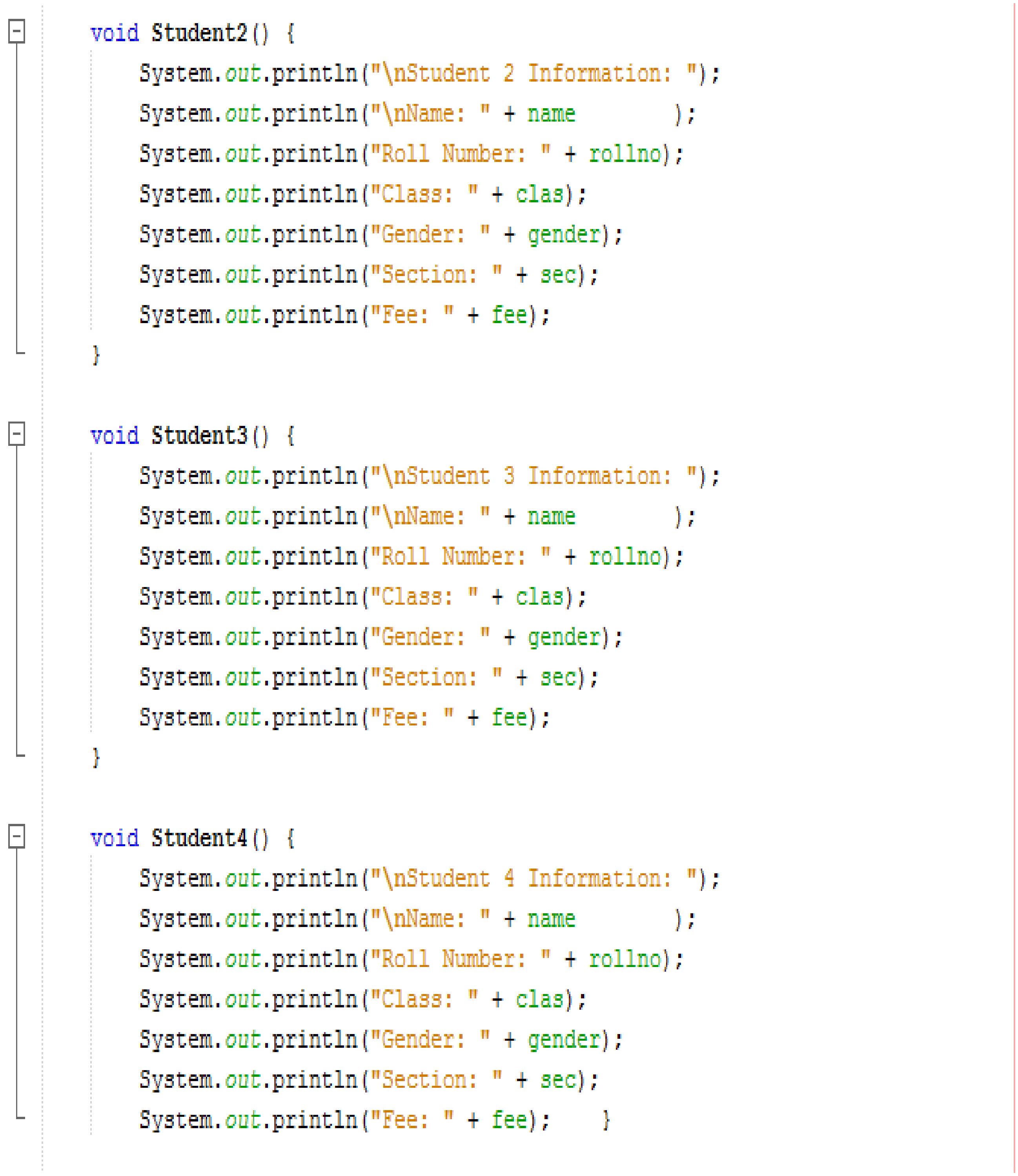
NAME OF STUDENT: Muhammad Owais Qadri **ID No: 519-2020**

Question 2(b): Make a school management system by **applying** object Oriented techniques.

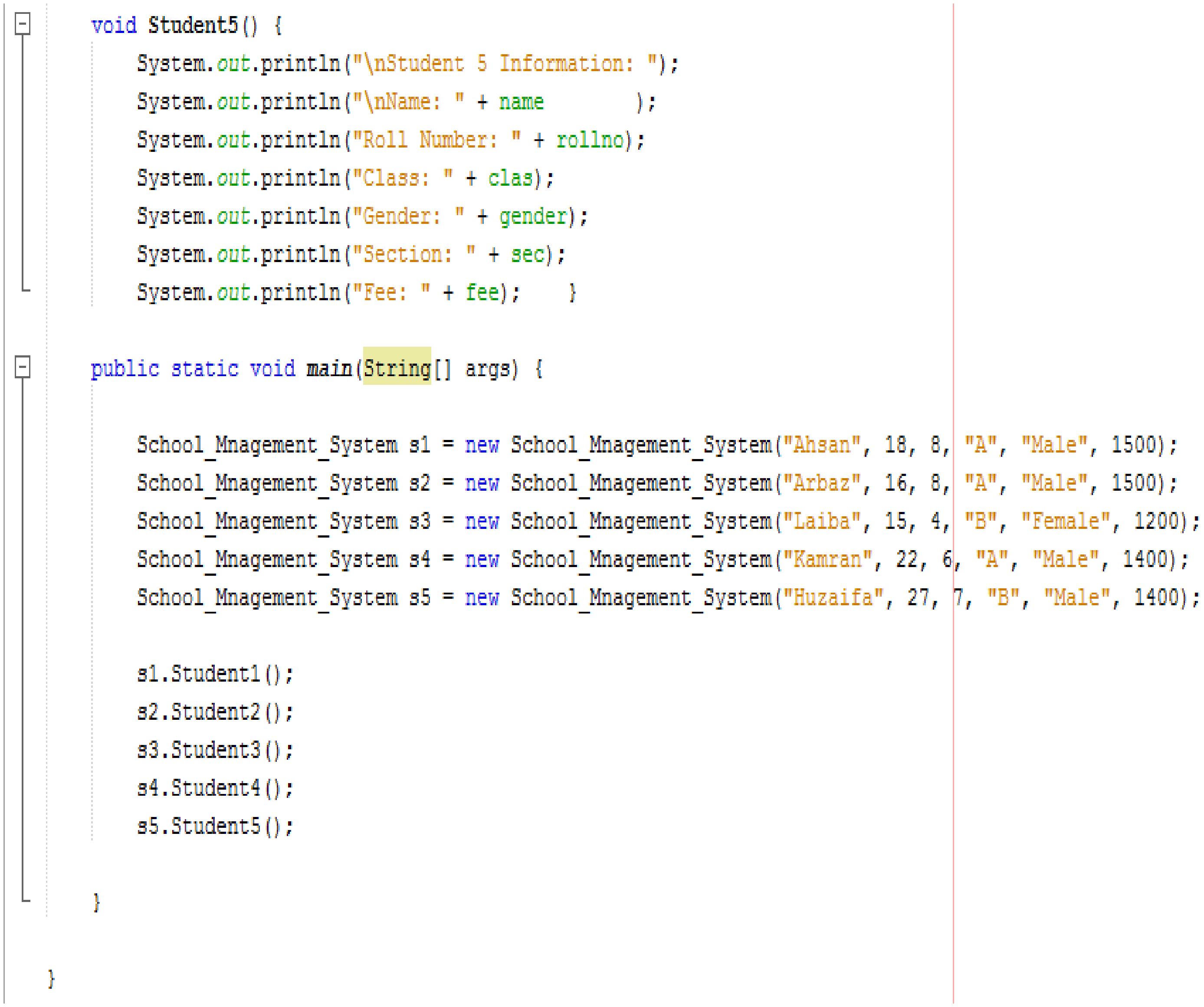
Code:



```
package school_mnagement_system;

public class School_Mnagement_System {
    String name;
    int rollno;
    String gender;
    int clas;
    String sec;
    float fee;

    School_Mnagement_System(String name, int rollno, int clas, String sec, String gender, float fee) {
        this.name = name;
        this.clas = clas;
        this.gender = gender;
        this.rollno = rollno;
        this.sec = sec;
        this.fee = fee;
    }

    void Student1() {
        System.out.println("\t\t\t\t\t\tSCHOOL MANAGEMENT SYSTEM");
        System.out.println("\nStudent 1 Information: ");
        System.out.println("\nName: " + name        );
        System.out.println("Roll Number: " + rollno);
        System.out.println("Class: " + clas);
        System.out.println("Gender: " + gender);
        System.out.println("Section: " + sec);
        System.out.println("Fee: " + fee);

    }
```

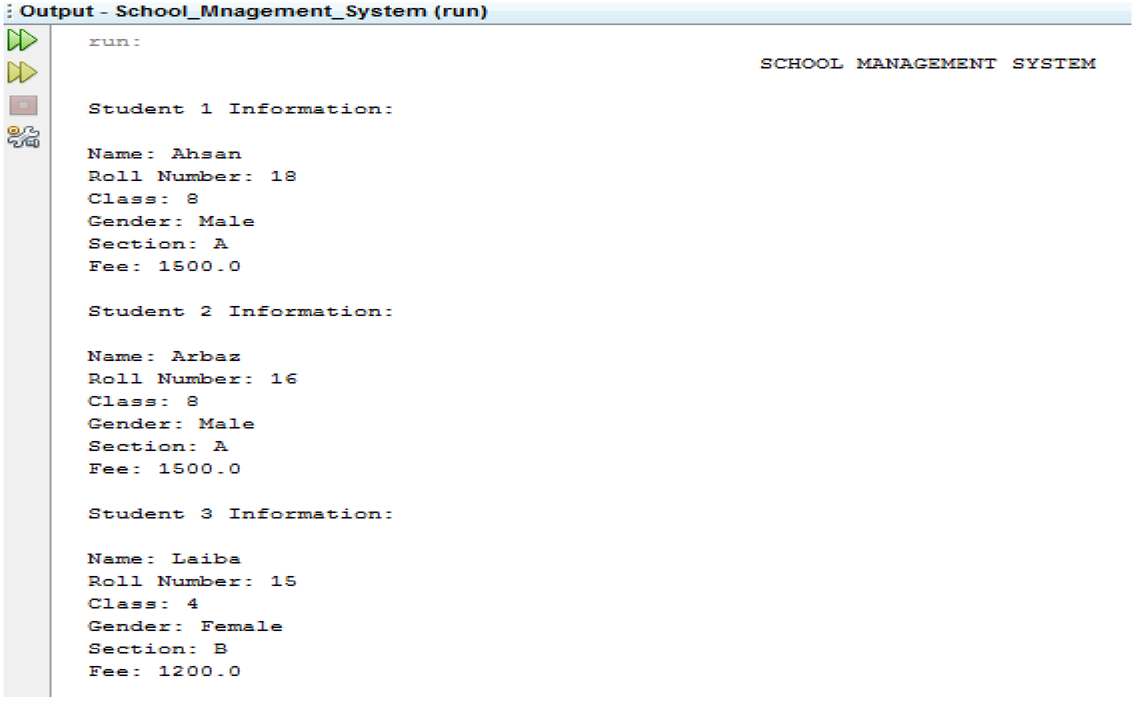


```
void Student2() {
    System.out.println("\nStudent 2 Information: ");
    System.out.println("\nName: " + name        );
    System.out.println("Roll Number: " + rollno);
    System.out.println("Class: " + clas);
    System.out.println("Gender: " + gender);
    System.out.println("Section: " + sec);
    System.out.println("Fee: " + fee);
}

void Student3() {
    System.out.println("\nStudent 3 Information: ");
    System.out.println("\nName: " + name        );
    System.out.println("Roll Number: " + rollno);
    System.out.println("Class: " + clas);
    System.out.println("Gender: " + gender);
    System.out.println("Section: " + sec);
    System.out.println("Fee: " + fee);
}

void Student4() {
    System.out.println("\nStudent 4 Information: ");
    System.out.println("\nName: " + name        );
    System.out.println("Roll Number: " + rollno);
    System.out.println("Class: " + clas);
    System.out.println("Gender: " + gender);
    System.out.println("Section: " + sec);
    System.out.println("Fee: " + fee);    }
```

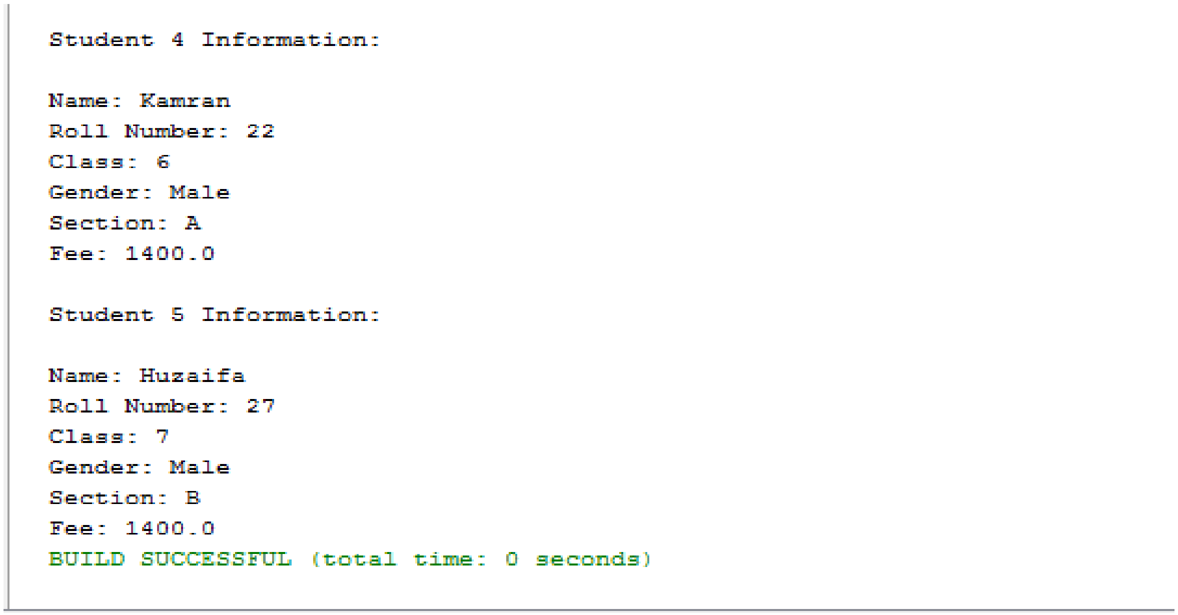


```
    void Student5() {
        System.out.println("\nStudent 5 Information: ");
        System.out.println("\nName: " + name        );
        System.out.println("Roll Number: " + rollno);
        System.out.println("Class: " + clas);
        System.out.println("Gender: " + gender);
        System.out.println("Section: " + sec);
        System.out.println("Fee: " + fee);    }

    public static void main(String[] args) {

        School_Mnagement_System s1 = new School_Mnagement_System("Ahsan", 18, 8, "A", "Male", 1500);
        School_Mnagement_System s2 = new School_Mnagement_System("Arbaz", 16, 8, "A", "Male", 1500);
        School_Mnagement_System s3 = new School_Mnagement_System("Laiba", 15, 4, "B", "Female", 1200);
        School_Mnagement_System s4 = new School_Mnagement_System("Kamran", 22, 6, "A", "Male", 1400);
        School_Mnagement_System s5 = new School_Mnagement_System("Huzaifa", 27, 7, "B", "Male", 1400);

        s1.Student1();
        s2.Student2();
        s3.Student3();
        s4.Student4();
        s5.Student5();

    }

}
```

Output:



```
run:
                                                        SCHOOL MANAGEMENT SYSTEM

Student 1 Information:

Name: Ahsan
Roll Number: 18
Class: 8
Gender: Male
Section: A
Fee: 1500.0

Student 2 Information:

Name: Arbaz
Roll Number: 16
Class: 8
Gender: Male
Section: A
Fee: 1500.0

Student 3 Information:

Name: Laiba
Roll Number: 15
Class: 4
Gender: Female
Section: B
Fee: 1200.0
```



```
Student 4 Information:

Name: Kamran
Roll Number: 22
Class: 6
Gender: Male
Section: A
Fee: 1400.0

Student 5 Information:

Name: Huzaifa
Roll Number: 27
Class: 7
Gender: Male
Section: B
Fee: 1400.0
BUILD SUCCESSFUL (total time: 0 seconds)
```