Harsh Kotadiya
22070126040
AIML-A2

Assignment-6-Part 1

Main.java

```java
// Main.java

package ASS_6_part_1;

import java.util.Scanner;

public class Main {
private static final Scanner scanner = new Scanner(System.in);

public static void main(String[] args) {
Interface_STK fixedStack = new Fixed_stk();
Interface_STK growableStack = new Growable_stk();

int choice;
do {
System.out.println("1. Push to Fixed Stack");
System.out.println("2. Pop from Fixed Stack");
System.out.println("3. Display Fixed Stack");
System.out.println("4. Push to Growable Stack");
System.out.println("5. Pop from Growable Stack");
System.out.println("6. Display Growable Stack");
System.out.println("0. Exit");

System.out.print("Enter your choice: ");
choice = scanner.nextInt();

switch (choice) {
case 1:
System.out.print("Enter element to push into Fixed Stack: ");
fixedStack.push(scanner.nextInt());
break;
case 2:
System.out.println("Popped element from Fixed Stack: " + fixedStack.pop());
break;
case 3:
System.out.println("Fixed Stack:");
fixedStack.displayStack();
break;
case 4:
System.out.print("Enter element to push into Growable Stack: ");
growableStack.push(scanner.nextInt());
break;
case 5:
System.out.println("Popped element from Growable Stack: " + growableStack.pop());
break;
case 6:
System.out.println("Growable Stack:");
growableStack.displayStack();
break;
case 0:
System.out.println("Exiting...");
break;
default:
System.out.println("Invalid choice. Please enter a valid option.");
}
} while (choice != 0);
}
}
```

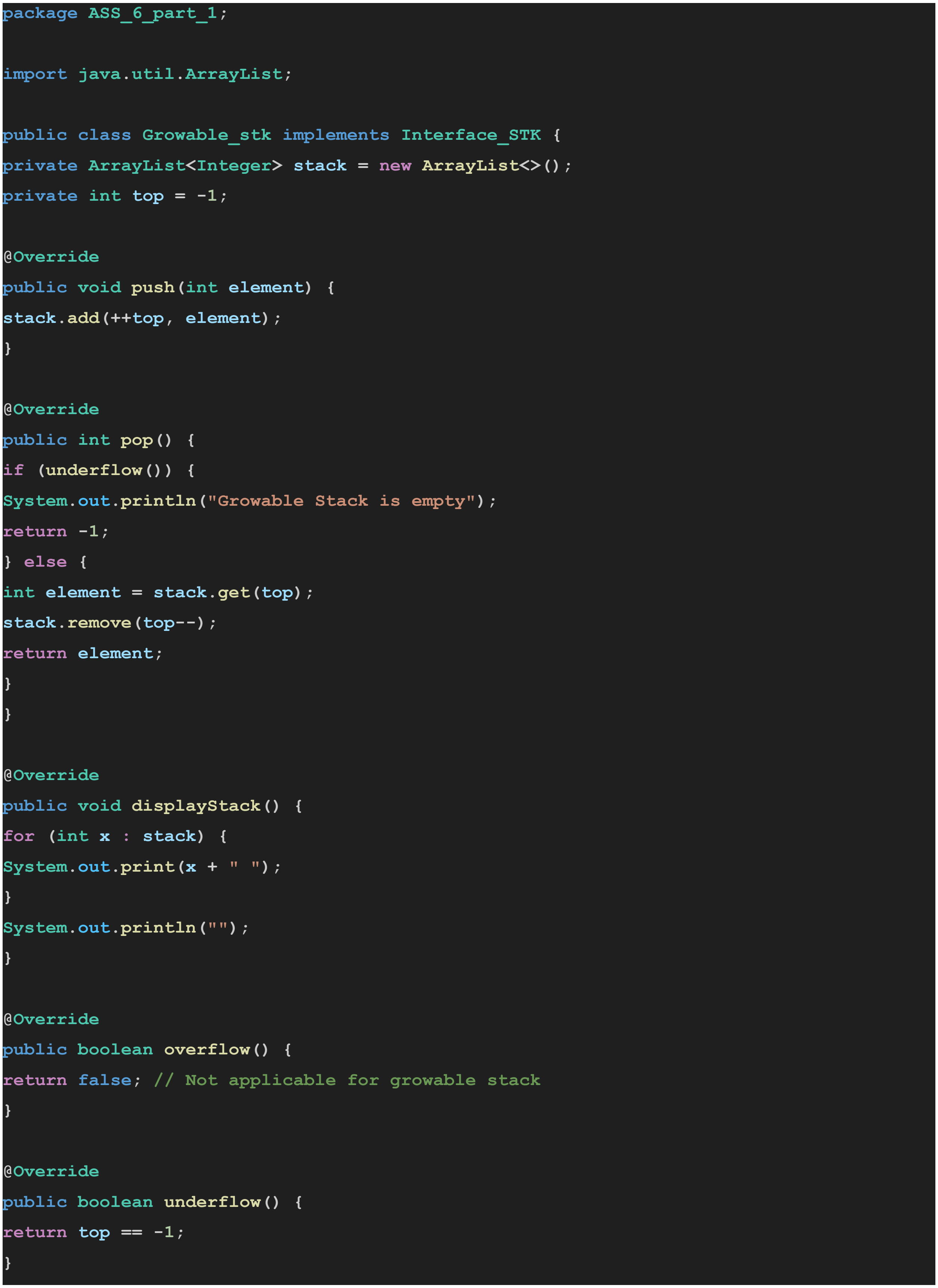
Fixed_stk.java

```
// Fixed_stk.java

package ASS_6_part_1;

public class Fixed_stk implements Interface_STK {
private static final int STACK_SIZE = 5;
private int stack[] = new int[STACK_SIZE];
private int top = -1;

@Override
public void push(int element) {
if (overflow()) {
```

```
System.out.println("Fixed Stack is full");
} else {
stack[++top] = element;
}
}

@Override
public int pop() {
if (underflow()) {
System.out.println("Fixed Stack is empty");
return -1;
} else {
return stack[top--];
}
}

@Override
public void displayStack() {
if (underflow()) {
System.out.println("Fixed Stack is empty");
}
for (int i = 0; i <= top; i++) {
System.out.println(stack[i]);
}
}

@Override
public boolean overflow() {
return top == (stack.length - 1);
}

@Override
public boolean underflow() {
return top == -1;
}
}
```

Growable_stk.java

```
// Growable_stk.java
```

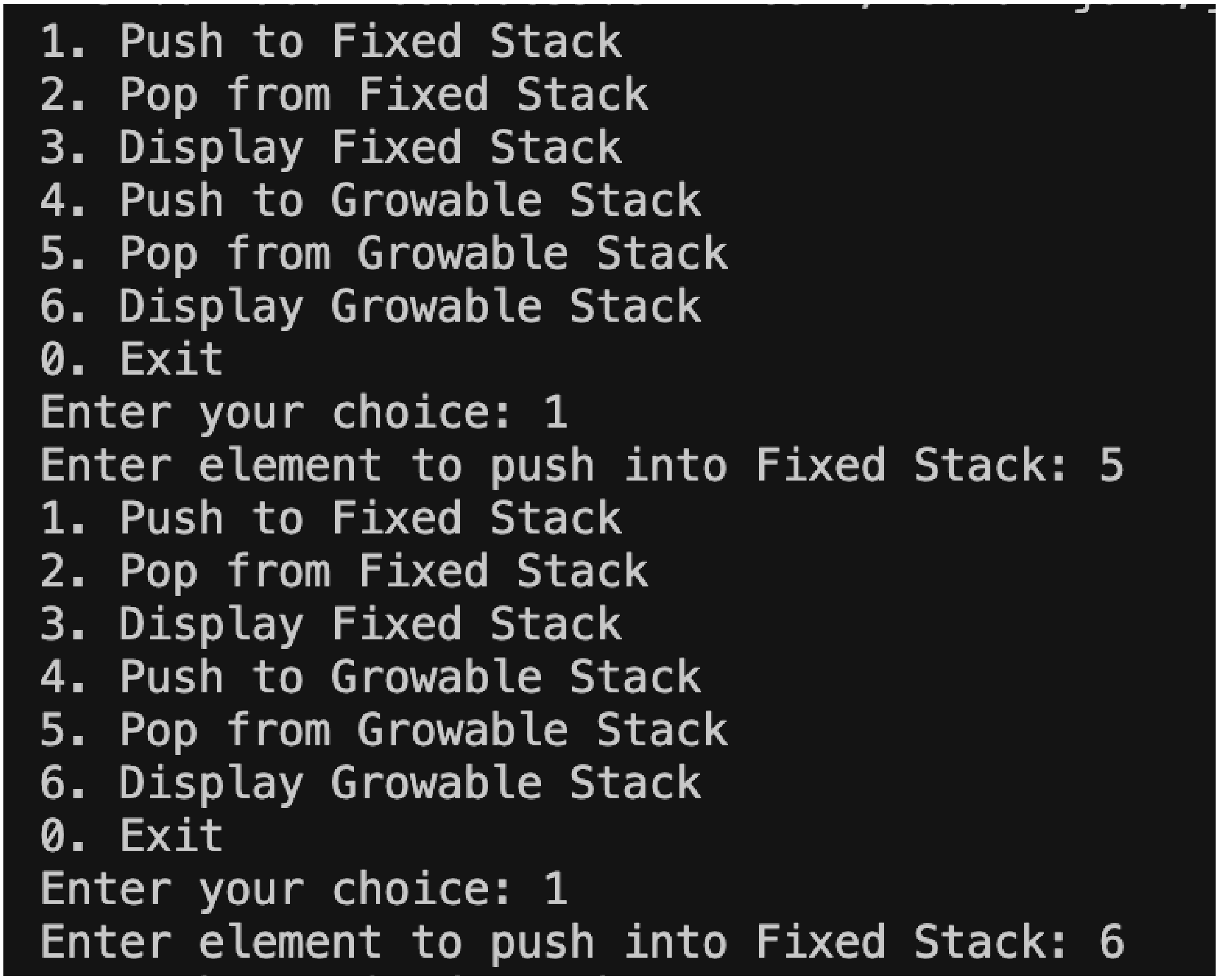


```
package ASS_6_part_1;

import java.util.ArrayList;

public class Growable_stk implements Interface_STK {
private ArrayList<Integer> stack = new ArrayList<>();
private int top = -1;

@Override
public void push(int element) {
stack.add(++top, element);
}

@Override
public int pop() {
if (underflow()) {
System.out.println("Growable Stack is empty");
return -1;
} else {
int element = stack.get(top);
stack.remove(top--);
return element;
}
}

@Override
public void displayStack() {
for (int x : stack) {
System.out.print(x + " ");
}
System.out.println("");
}

@Override
public boolean overflow() {
return false; // Not applicable for growable stack
}

@Override
public boolean underflow() {
return top == -1;
}
}
```

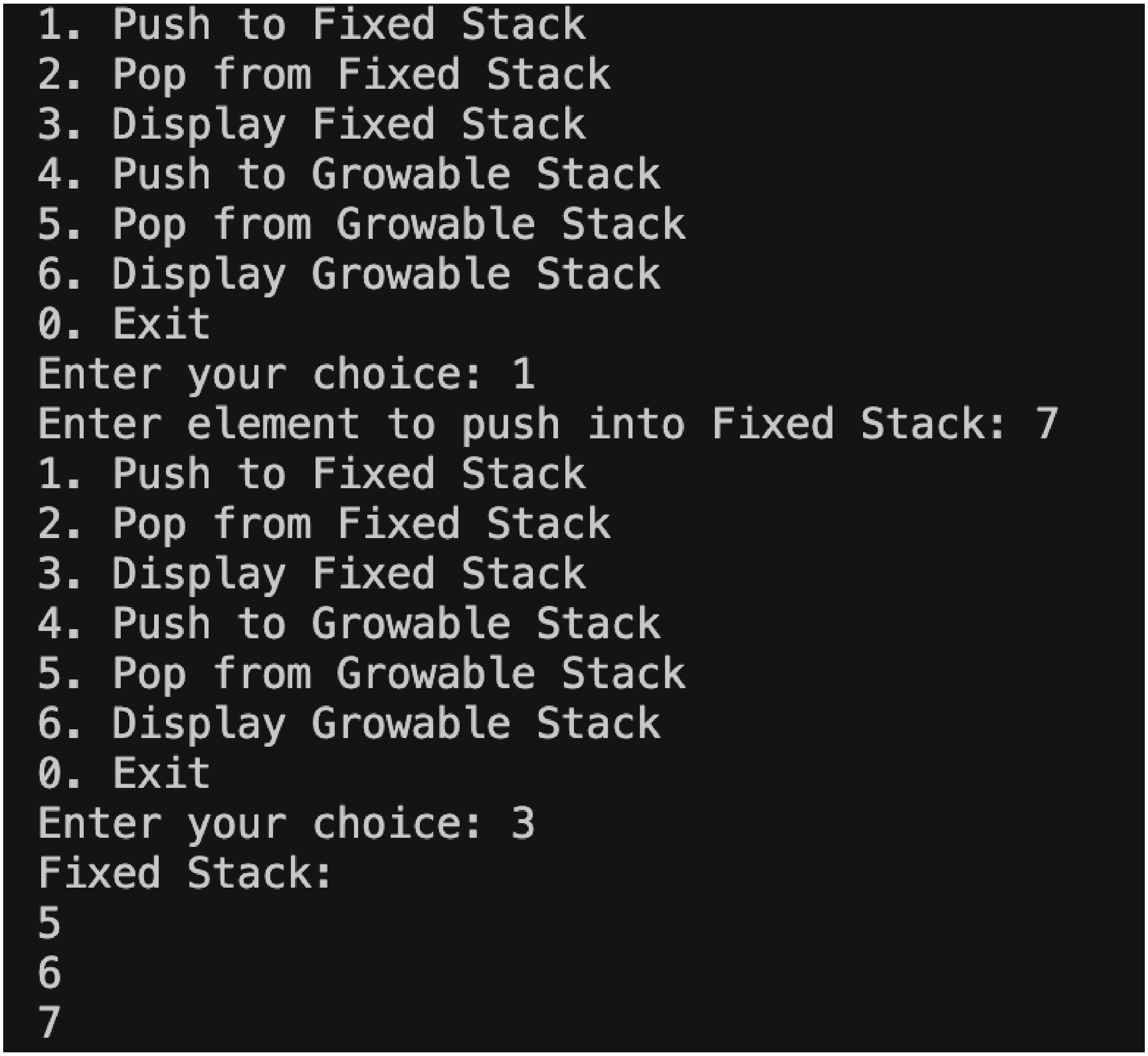
Interface_STK.java

```java
// Interface_STK.java

package ASS_6_part_1;

public interface Interface_STK {
void push(int element);

int pop();

void displayStack();

boolean overflow();

boolean underflow();
}
```

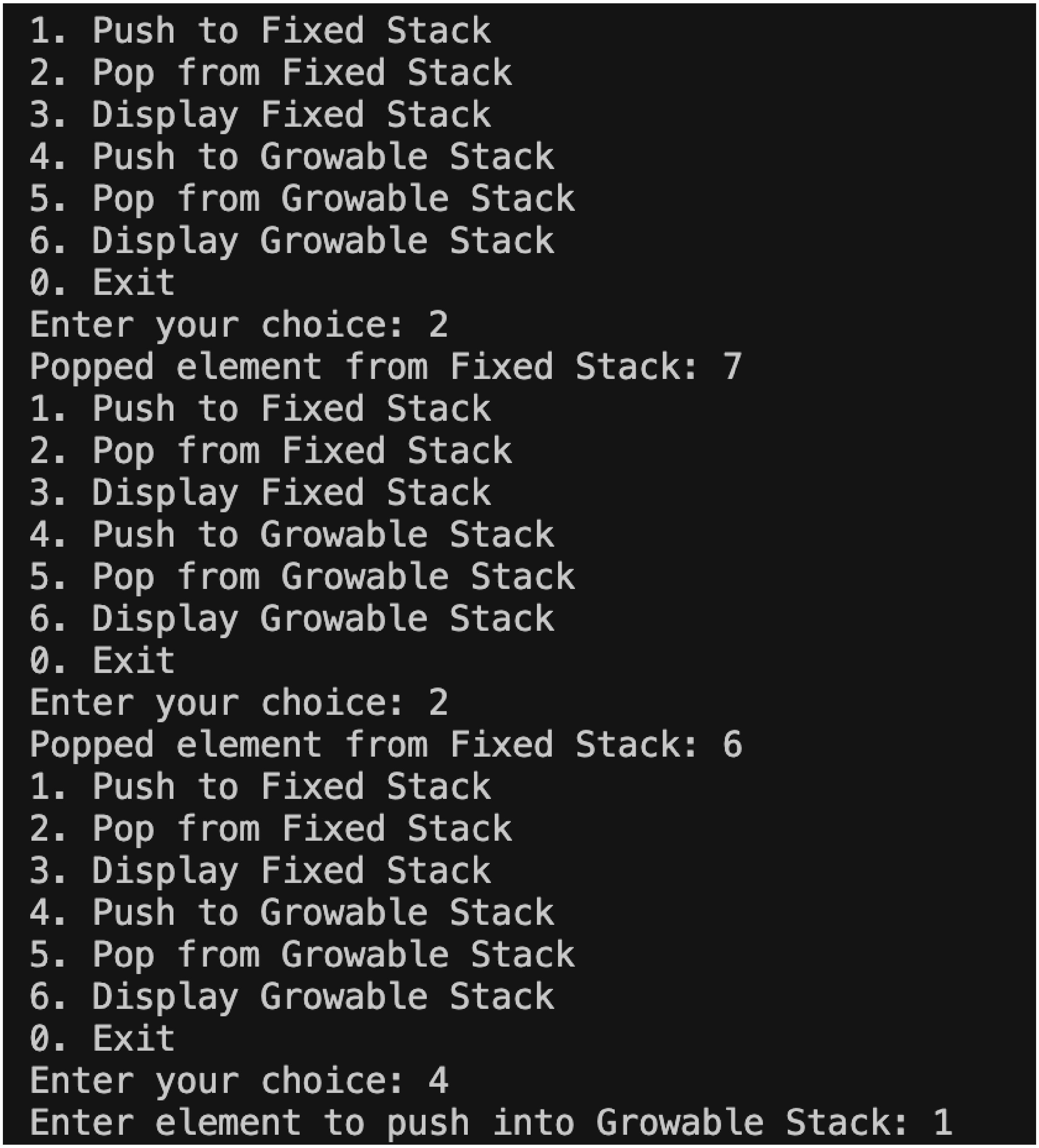
Outputs of part-2



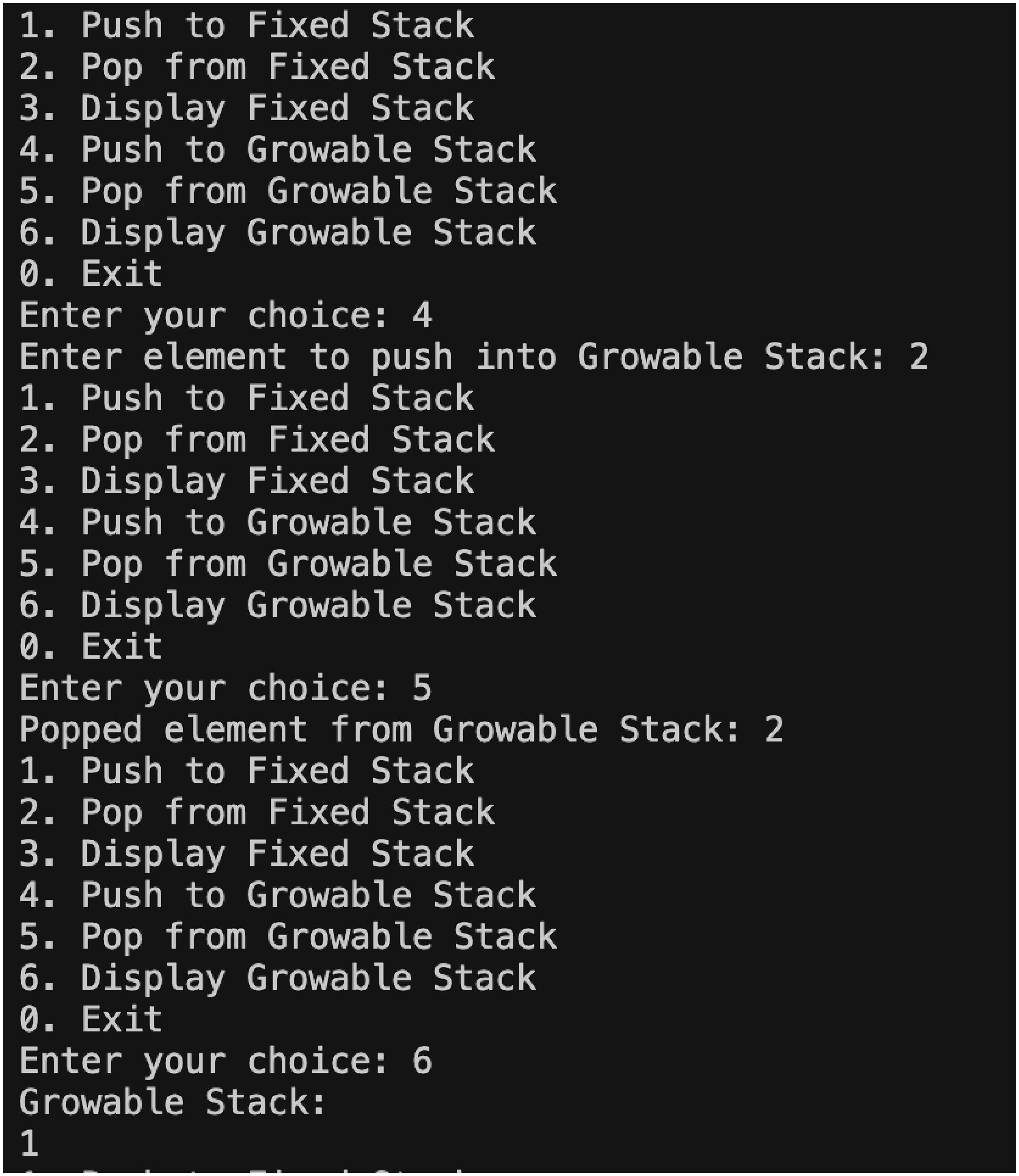
```
1. Push to Fixed Stack
2. Pop from Fixed Stack
3. Display Fixed Stack
4. Push to Growable Stack
5. Pop from Growable Stack
6. Display Growable Stack
0. Exit
Enter your choice: 1
Enter element to push into Fixed Stack: 5
1. Push to Fixed Stack
2. Pop from Fixed Stack
3. Display Fixed Stack
4. Push to Growable Stack
5. Pop from Growable Stack
6. Display Growable Stack
0. Exit
Enter your choice: 1
Enter element to push into Fixed Stack: 6
```



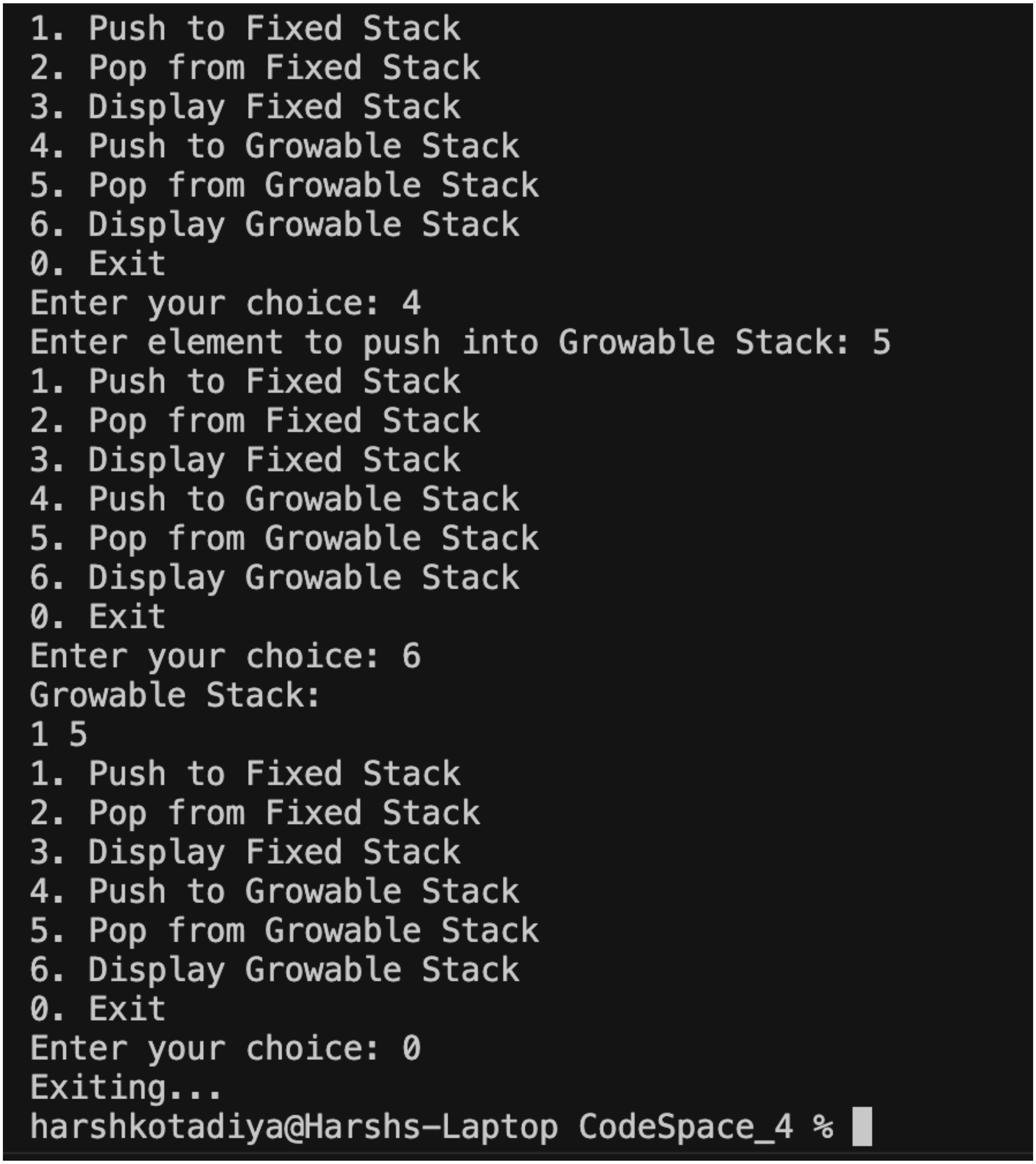
```
1. Push to Fixed Stack
2. Pop from Fixed Stack
3. Display Fixed Stack
4. Push to Growable Stack
5. Pop from Growable Stack
6. Display Growable Stack
0. Exit
Enter your choice: 1
Enter element to push into Fixed Stack: 7
1. Push to Fixed Stack
2. Pop from Fixed Stack
3. Display Fixed Stack
4. Push to Growable Stack
5. Pop from Growable Stack
6. Display Growable Stack
0. Exit
Enter your choice: 3
Fixed Stack:
5
6
7
```



```
1. Push to Fixed Stack
2. Pop from Fixed Stack
3. Display Fixed Stack
4. Push to Growable Stack
5. Pop from Growable Stack
6. Display Growable Stack
0. Exit
Enter your choice: 2
Popped element from Fixed Stack: 7
1. Push to Fixed Stack
2. Pop from Fixed Stack
3. Display Fixed Stack
4. Push to Growable Stack
5. Pop from Growable Stack
6. Display Growable Stack
0. Exit
Enter your choice: 2
Popped element from Fixed Stack: 6
1. Push to Fixed Stack
2. Pop from Fixed Stack
3. Display Fixed Stack
4. Push to Growable Stack
5. Pop from Growable Stack
6. Display Growable Stack
0. Exit
Enter your choice: 4
Enter element to push into Growable Stack: 1
```



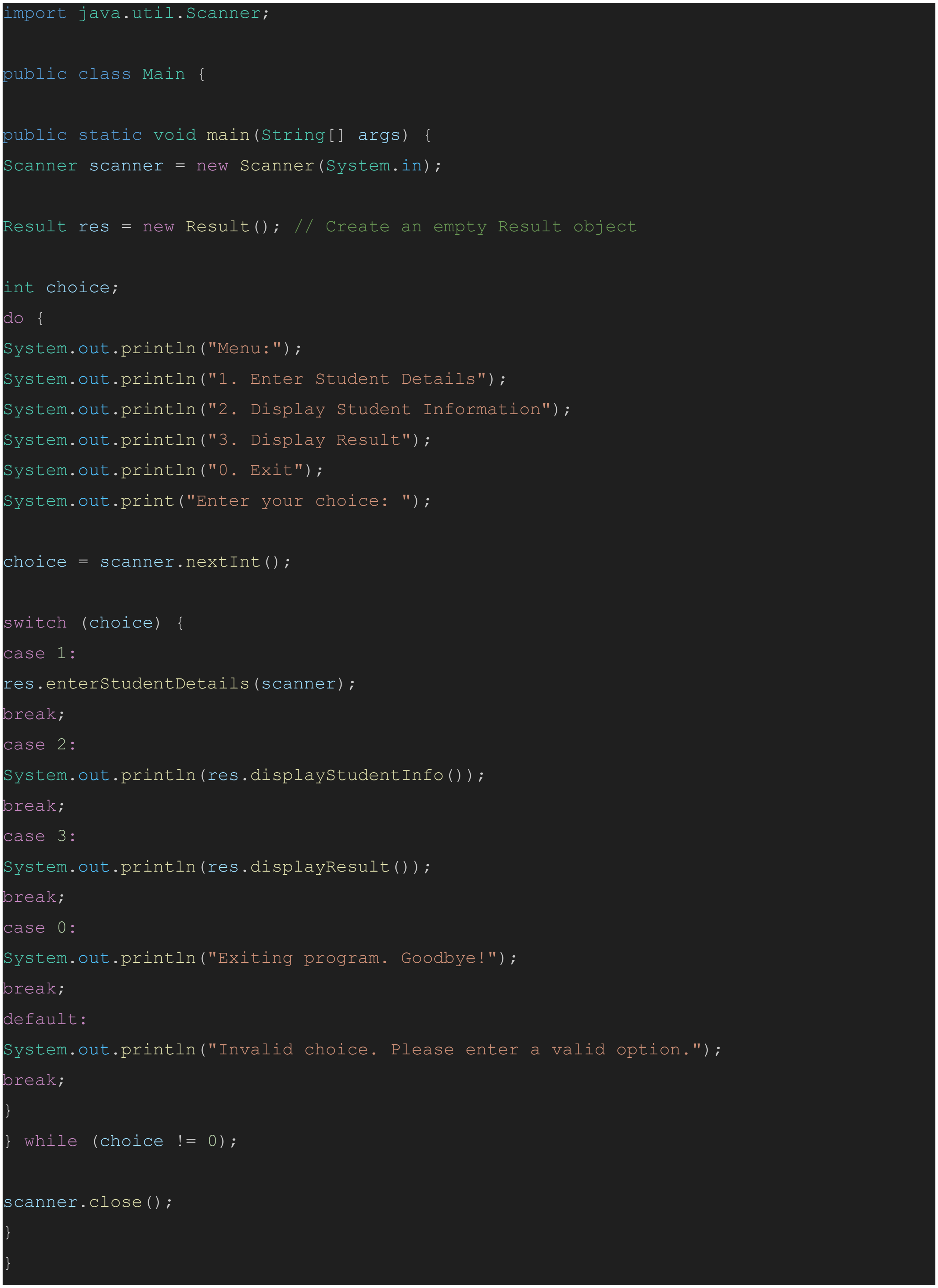
```
1. Push to Fixed Stack
2. Pop from Fixed Stack
3. Display Fixed Stack
4. Push to Growable Stack
5. Pop from Growable Stack
6. Display Growable Stack
0. Exit
Enter your choice: 4
Enter element to push into Growable Stack: 2
1. Push to Fixed Stack
2. Pop from Fixed Stack
3. Display Fixed Stack
4. Push to Growable Stack
5. Pop from Growable Stack
6. Display Growable Stack
0. Exit
Enter your choice: 5
Popped element from Growable Stack: 2
1. Push to Fixed Stack
2. Pop from Fixed Stack
3. Display Fixed Stack
4. Push to Growable Stack
5. Pop from Growable Stack
6. Display Growable Stack
0. Exit
Enter your choice: 6
Growable Stack:
1
```



```
1. Push to Fixed Stack
2. Pop from Fixed Stack
3. Display Fixed Stack
4. Push to Growable Stack
5. Pop from Growable Stack
6. Display Growable Stack
0. Exit
Enter your choice: 4
Enter element to push into Growable Stack: 5
1. Push to Fixed Stack
2. Pop from Fixed Stack
3. Display Fixed Stack
4. Push to Growable Stack
5. Pop from Growable Stack
6. Display Growable Stack
0. Exit
Enter your choice: 6
Growable Stack:
1 5
1. Push to Fixed Stack
2. Pop from Fixed Stack
3. Display Fixed Stack
4. Push to Growable Stack
5. Pop from Growable Stack
6. Display Growable Stack
0. Exit
Enter your choice: 0
Exiting...
harshkotadiya@Harshs-Laptop CodeSpace_4 %
```

Assignment-6-Part 2

Main.java



```
import java.util.Scanner;

public class Main {

public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);

Result res = new Result(); // Create an empty Result object

int choice;
do {
System.out.println("Menu:");
System.out.println("1. Enter Student Details");
System.out.println("2. Display Student Information");
System.out.println("3. Display Result");
System.out.println("0. Exit");
System.out.print("Enter your choice: ");

choice = scanner.nextInt();

switch (choice) {
case 1:
res.enterStudentDetails(scanner);
break;
case 2:
System.out.println(res.displayStudentInfo());
break;
case 3:
System.out.println(res.displayResult());
break;
case 0:
System.out.println("Exiting program. Goodbye!");
break;
default:
System.out.println("Invalid choice. Please enter a valid option.");
break;
}
} while (choice != 0);

scanner.close();
}
}
```

Exam.java

```java
public interface Exam {
void Percent_cal();
}
```

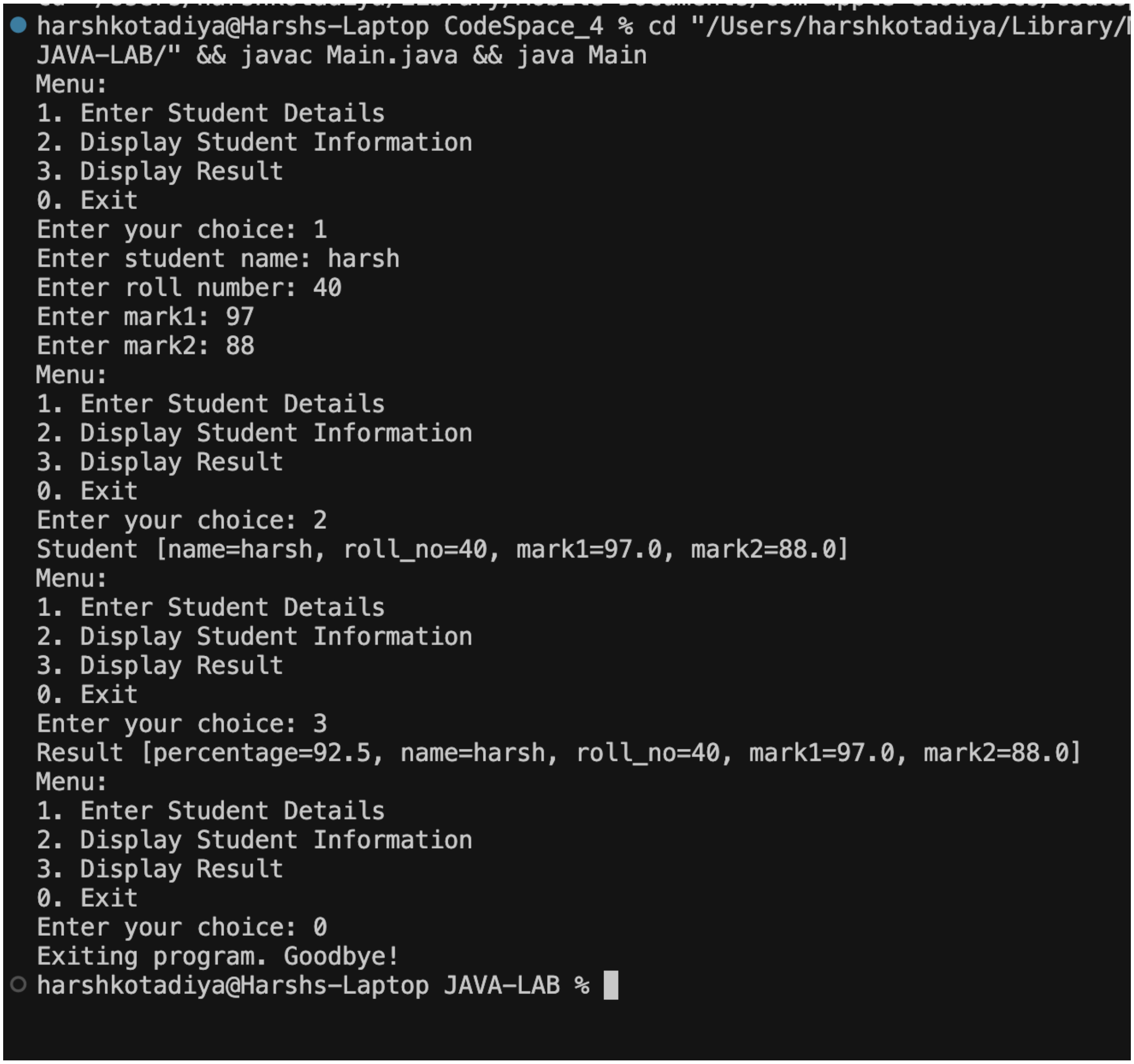
Result.java

```java
import java.util.Scanner;

public class Result extends Student implements Exam {
double percentage;

public Result() {
super("", 0, 0, 0); // Initialize with empty/default values
}

@Override
public void Percent_cal() {
this.percentage = (mark1 + mark2) / 2;
}

public void enterStudentDetails(Scanner scanner) {
System.out.print("Enter student name: ");
setName(scanner.next());
System.out.print("Enter roll number: ");
setRoll_no(scanner.nextInt());
System.out.print("Enter mark1: ");
setMark1(scanner.nextDouble());
System.out.print("Enter mark2: ");
setMark2(scanner.nextDouble());
}

public String displayStudentInfo() {
return super.toString(); // Using toString from Student class
}
public String displayResult() {
Percent_cal();
return toString();
}

@Override
public String toString() {
return "Result [percentage=" + percentage + ", name=" + name + ", roll_no=" + roll_no
+ ", mark1=" + mark1
+ ", mark2=" + mark2 + "]";
}
}
```

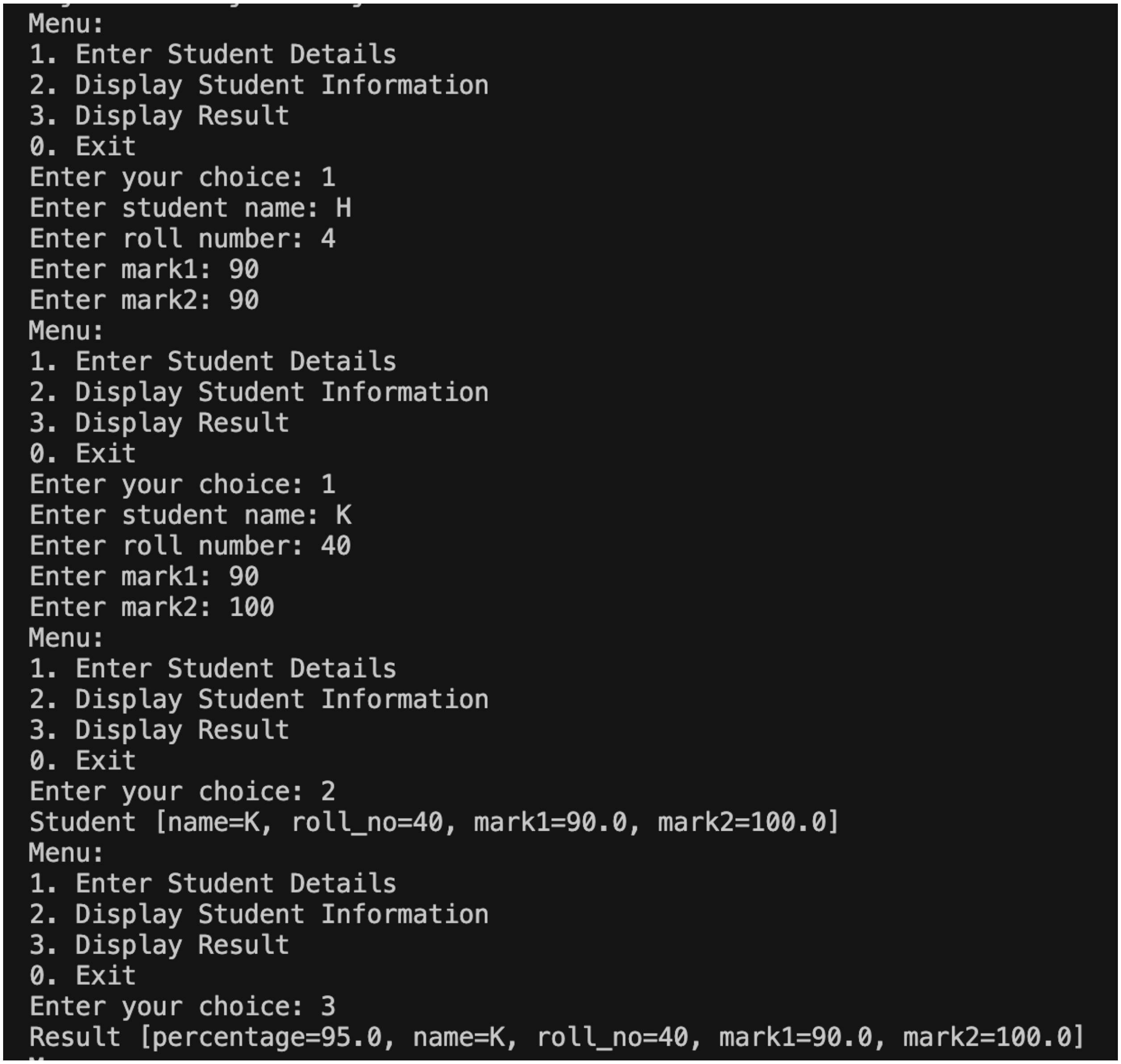
Student.java

```
public class Student {
String name;
int roll_no;
double mark1, mark2;

public Student(String name, int roll_no, double mark1, double mark2) {
this.name = name;
this.roll_no = roll_no;
this.mark1 = mark1;
this.mark2 = mark2;
}

// Getter methods
public String getName() {
return name;
}

public int getRoll_no() {
return roll_no;
}

public double getMark1() {
return mark1;
}
public double getMark2() {
return mark2;
}

// Setter methods
public void setName(String name) {
this.name = name;
}

public void setRoll_no(int roll_no) {
this.roll_no = roll_no;
}

public void setMark1(double mark1) {
this.mark1 = mark1;
}

public void setMark2(double mark2) {
this.mark2 = mark2;
}

@Override
public String toString() {
return "Student [name=" + name + ", roll_no=" + roll_no + ", mark1=" + mark1 + ",
mark2=" + mark2 + "]";
}
}
```

Outputs of part-2



```
● harshkotadiya@Harshs-Laptop CodeSpace_4 % cd "/Users/harshkotadiya/Library/M
  JAVA-LAB/" && javac Main.java && java Main
  Menu:
  1. Enter Student Details
  2. Display Student Information
  3. Display Result
  0. Exit
  Enter your choice: 1
  Enter student name: harsh
  Enter roll number: 40
  Enter mark1: 97
  Enter mark2: 88
  Menu:
  1. Enter Student Details
  2. Display Student Information
  3. Display Result
  0. Exit
  Enter your choice: 2
  Student [name=harsh, roll_no=40, mark1=97.0, mark2=88.0]
  Menu:
  1. Enter Student Details
  2. Display Student Information
  3. Display Result
  0. Exit
  Enter your choice: 3
  Result [percentage=92.5, name=harsh, roll_no=40, mark1=97.0, mark2=88.0]
  Menu:
  1. Enter Student Details
  2. Display Student Information
  3. Display Result
  0. Exit
  Enter your choice: 0
  Exiting program. Goodbye!
○ harshkotadiya@Harshs-Laptop JAVA-LAB %
```



```
Menu:
1. Enter Student Details
2. Display Student Information
3. Display Result
0. Exit
Enter your choice: 1
Enter student name: H
Enter roll number: 4
Enter mark1: 90
Enter mark2: 90
Menu:
1. Enter Student Details
2. Display Student Information
3. Display Result
0. Exit
Enter your choice: 1
Enter student name: K
Enter roll number: 40
Enter mark1: 90
Enter mark2: 100
Menu:
1. Enter Student Details
2. Display Student Information
3. Display Result
0. Exit
Enter your choice: 2
Student [name=K, roll_no=40, mark1=90.0, mark2=100.0]
Menu:
1. Enter Student Details
2. Display Student Information
3. Display Result
0. Exit
Enter your choice: 3
Result [percentage=95.0, name=K, roll_no=40, mark1=90.0, mark2=100.0]
```