OOP LAB

Week 7 Assignment Submission

Swamiraju Satya Praveen Varma

200905044

Batch B1

10

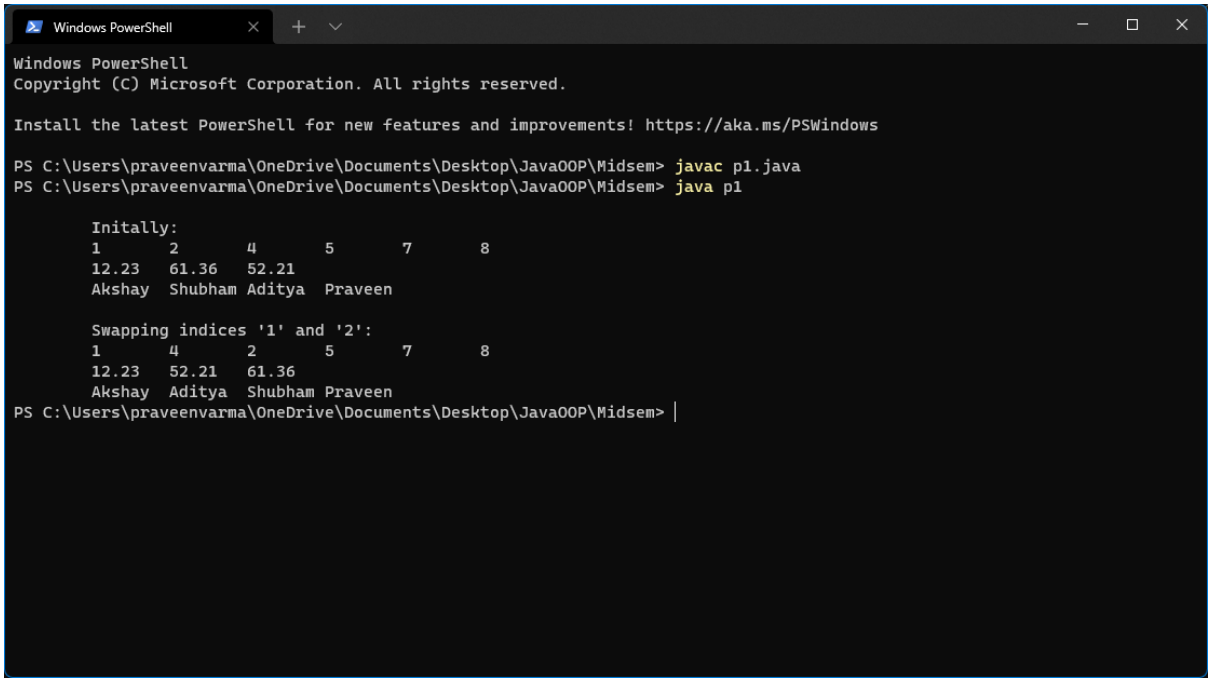
Question 1:- Write a generic method to exchange the positions of two different elements in an array.

Solution:-

```
class GArray <Type>
{
public static <Type> void exchange (Type [] arr, int a, int b)
{
Type temp = arr[a];
arr[a] = arr[b];
arr[b] = temp;
}
public static <Type> void display (Type [] arr)
{
for (Type obj: arr)
{
System.out.print("\t" + obj);
}
```

```
System.out.println();
}
}
public class p1
{
public static void main (String [] args)
{
Integer intArray[] = {1, 2, 4, 5, 7, 8};
Double doubleArray[] = {12.23, 61.36, 52.21};
String stringArray[] = {"Akshay", "Shubham", "Aditya", "Praveen"};
System.out.println("\n\tInitally: ");
GArray.display(intArray);
GArray.display(doubleArray);
GArray.display(stringArray);
System.out.println("\n\tSwapping indices '1' and '2': ");
GArray.exchange(intArray, 1, 2);
GArray.exchange(doubleArray, 1, 2);
GArray.exchange(stringArray, 1, 2);
GArray.display(intArray);
GArray.display(doubleArray);
GArray.display(stringArray);
}
}
```

Sample Input/output:



Question 2:- Define a simple generic stack class and show. The use of the generic class. For two different class types Student and Employee class objects.

Solution:-

```
import java.util.Scanner;


class Stack<Type> {
    private Type arr[];
    private int tos;

    public Stack(int n) {
        tos = -1;
        arr = (Type[]) new Object[n];
    }
```

```
public boolean isEmpty() {
   return (tos == -1);
}

public void push(Type item) {
   if (tos == arr.length - 1) {
      System.out.println("\n\tSTACK OVERFLOW!");
      return;
   }
   arr[++tos] = item;
}

public Type pop() {
   if (tos == -1) {
      System.out.println("\n\tSTACK UNDERFLOW!");
      return null;
   }
   return arr[tos--];
}

@Override
public String toString() {
   if (tos == -1) {
```

```
            return "STACK IS EMPTY!";
        }
        String str = "";
        for (int i = 0; i <= tos; ++i) {
            str += "\t" + arr[i];
        }
        return str;
    }
}

class Student {
    private String name;
    private double cgpa;

    public void input() {
        Scanner sc = new Scanner(System.in);
        System.out.print("\n\tEnter student name: ");
        name = sc.nextLine();
        System.out.print("\tEnter student cgpa: ");
        cgpa = sc.nextDouble();
    }

    @Override
    public String toString() {
```

```
        return "\n\t| STUDENT\n\t| NAME: " + name + "\n\t| CGPA: " +
cgpa
                + "\n";
    }
}

class Employee {
    private String name;
    private String idno;

    public void input() {
        Scanner sc = new Scanner(System.in);
        System.out.print("\n\tEnter employee name: ");
        name = sc.nextLine();
        System.out.print("\tEnter employee id: ");
        idno = sc.nextLine();
    }

    @Override
    public String toString() {
        return "\n\t| EMPLOYEE\n\t| NAME: " + name + "\n\t| IDNO: "
+ idno
                + "\n";
    }
```

```
}


public class p2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("\n\tEnter the size of the stacks: ");
        int n = sc.nextInt();
        Stack<Student> sstack = new Stack<>(n);
        Stack<Employee> estack = new Stack<>(n);
        int choice;
        do {
            System.out.print("\n\t1. Student\n\t2. Employee\n\tCHOICE:
");
            choice = sc.nextInt();
            if (choice < 1 || choice > 2) {
                System.out.println("Invalid Choice!");
                System.exit(0);
            }
            int stch;
            do {
                System.out.print("\n\t1. Push\n\t2. Pop\n\t3.
Display\n\tChoice: ");
                stch = sc.nextInt();
                if (stch < 1 || stch > 3) {
```

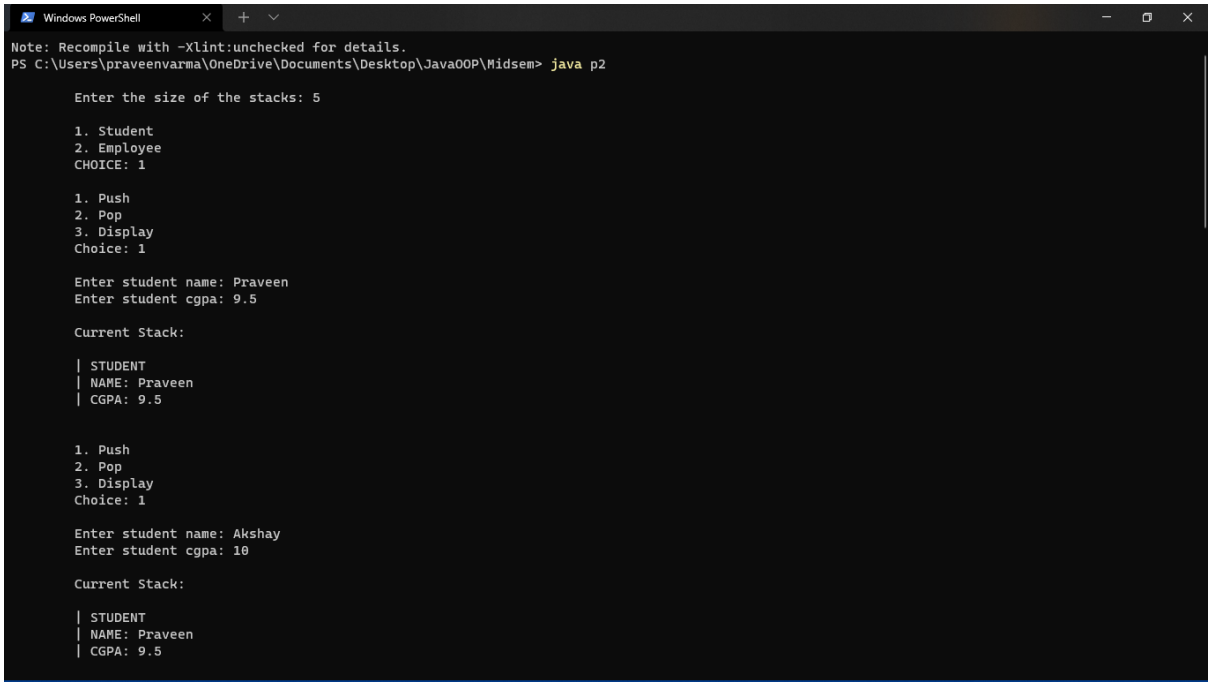
```
    break;
}
if (stch == 1) {
    if (choice == 1) {
        Student stud = new Student();
        stud.input();
        sstack.push(stud);
    } else {
        Employee empl = new Employee();
        empl.input();
        estack.push(empl);
    }
} else if (stch == 2) {
    if (choice == 1) {
        Student stud = sstack.pop();
        if (stud != null) {
            System.out.print("\nPopped: " + stud);
        }
    } else {
        Employee empl = estack.pop();
        if (empl != null) {
            System.out.print("\nPopped: " + empl);
        }
    }
```

```
            }
            if (choice == 1) {
               if (!sstack.isEmpty()) {
                  System.out.println("\n\tCurrent Stack: \n" +
                        sstack);
               }
            } else {
               if (!estack.isEmpty()) {
                  System.out.println("\n\tCurrent Stack: \n" +
                        estack);
               }
            }
         } while (stch >= 1 && stch <= 3);
      } while (choice == 1 || choice == 2);
   }
}
```

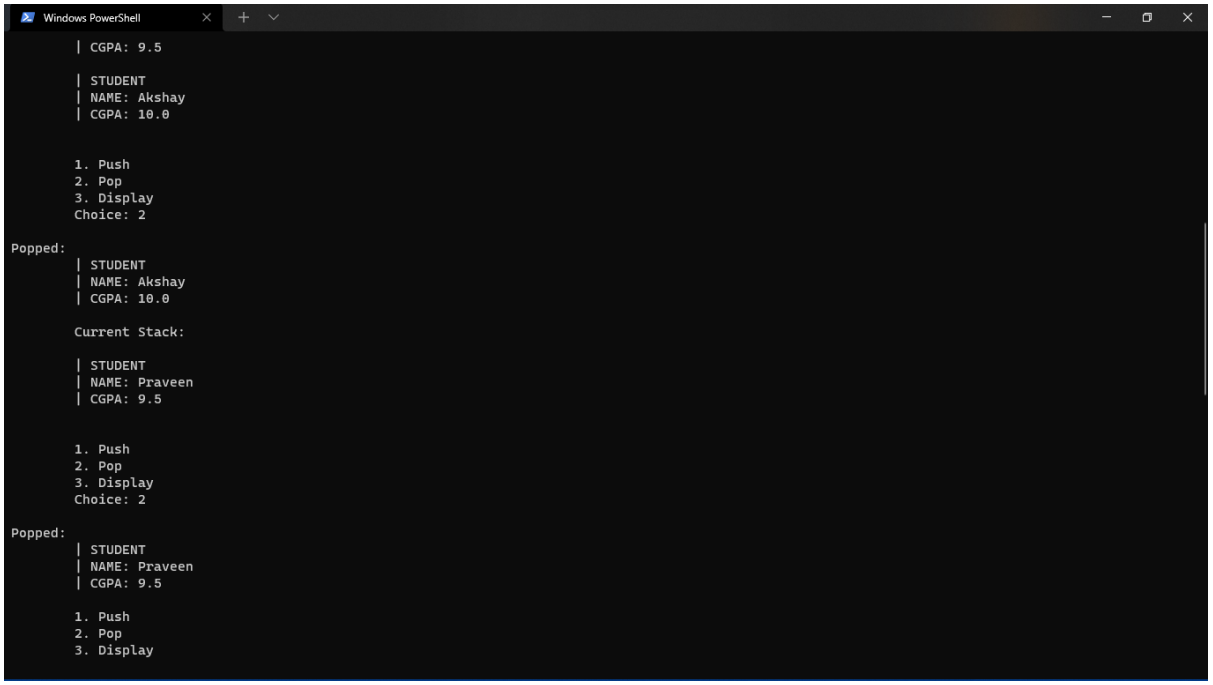
Sample input/output:

Choosing Student between Student and Employee

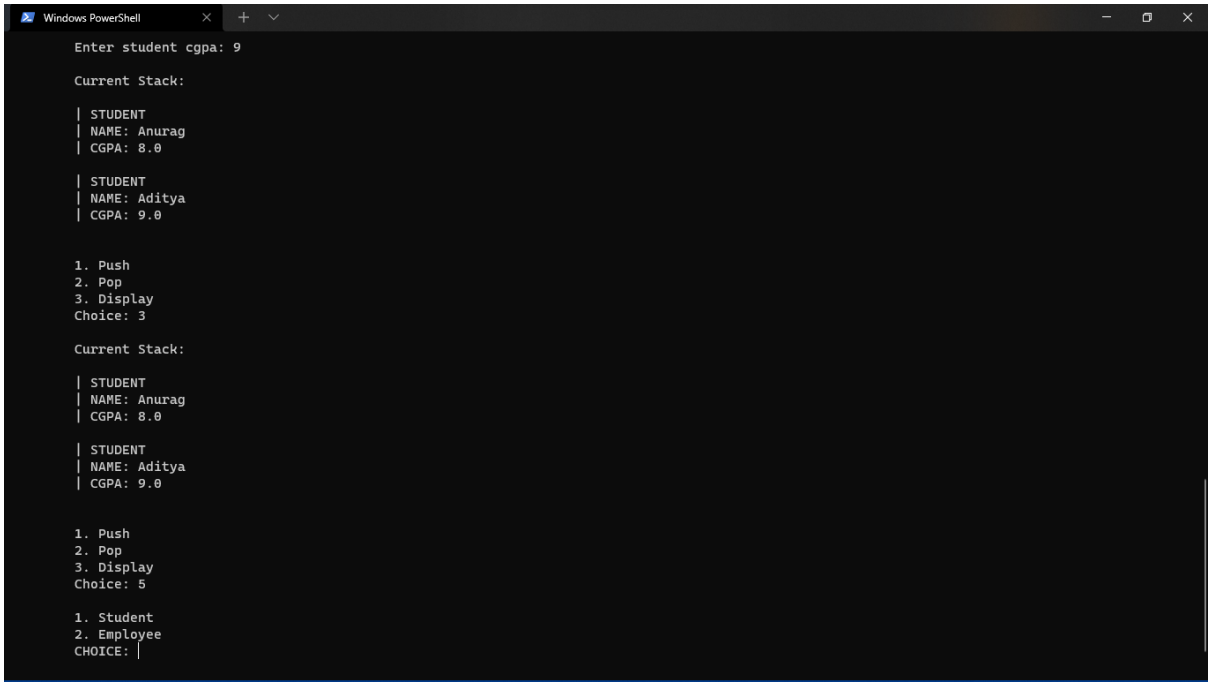
Pushing into Student Stack



Popping the Student stack

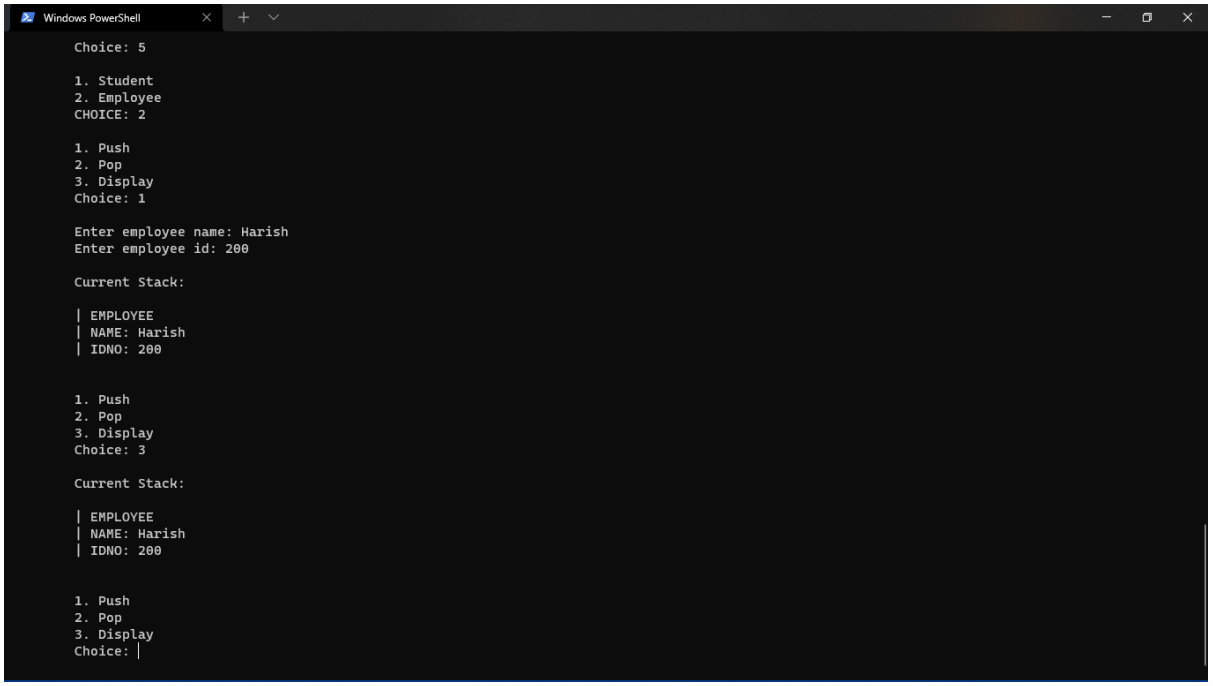


Display the current stack



Choosing Employee between Student and Employee

Pushing and displaying Stack



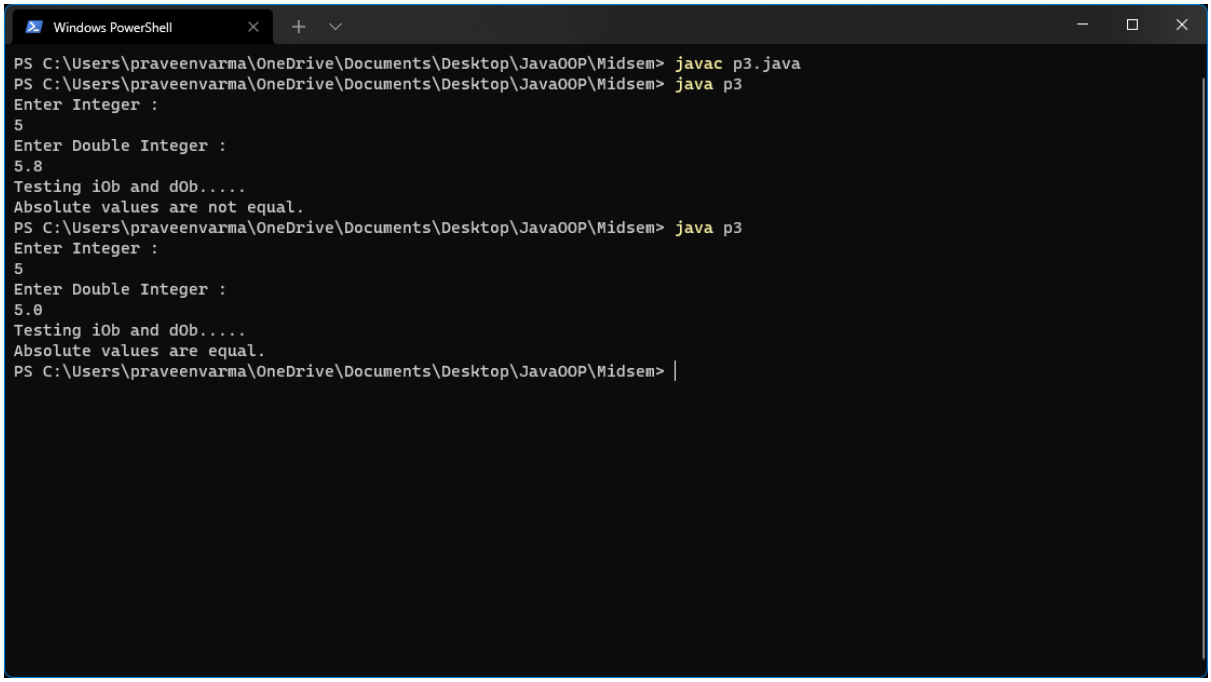
Question3:- Write a program to demonstrate the use of wildcard arguments.

Solution:-

```
import java.util.*;


class NumericFns<T extends Number> {
   T num;


   NumericFns(T n) {
      num = n;
   }


   double reciprocal() {
      return 1 / num.doubleValue();
   }


   double fraction() {
      return (num.doubleValue() - num.intValue());
   }


   boolean absEqual(NumericFns<?> ob) {
      if (Math.abs(num.doubleValue()) ==
Math.abs(ob.num.doubleValue())) {
         return true;
```

```
        }
        return false;
    }
}

class p3 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Integer : ");
        int i = sc.nextInt();
        System.out.println("Enter Double Integer : ");
        double d = sc.nextDouble();
        NumericFns<Integer> iOb = new NumericFns<Integer>(i);
        NumericFns<Double> dOb = new NumericFns<Double>(d);
        System.out.println("Testing iOb and dOb.....");
        if (iOb.absEqual(dOb)) {
            System.out.println("Absolute values are equal.");
        } else {
            System.out.println("Absolute values are not equal.");
        }
    }
}
```

Sample input/output:


Windows PowerShell
PS C:\Users\praveenvarma\OneDrive\Documents\Desktop\JavaOOP\Midsem> javac p3.java
PS C:\Users\praveenvarma\OneDrive\Documents\Desktop\JavaOOP\Midsem> java p3
Enter Integer :
5
Enter Double Integer :
5.8
Testing iOb and dOb.....
Absolute values are not equal.
PS C:\Users\praveenvarma\OneDrive\Documents\Desktop\JavaOOP\Midsem> java p3
Enter Integer :
5
Enter Double Integer :
5.0
Testing iOb and dOb.....
Absolute values are equal.
PS C:\Users\praveenvarma\OneDrive\Documents\Desktop\JavaOOP\Midsem>

THANK YOU!