CSE 1007

JAVA Programming

LAB ASSESSMENT - 5

NAME: Vibhu Kumar Singh
REG. NO: 19BCE0215
TEACHER: Jaisankar N.

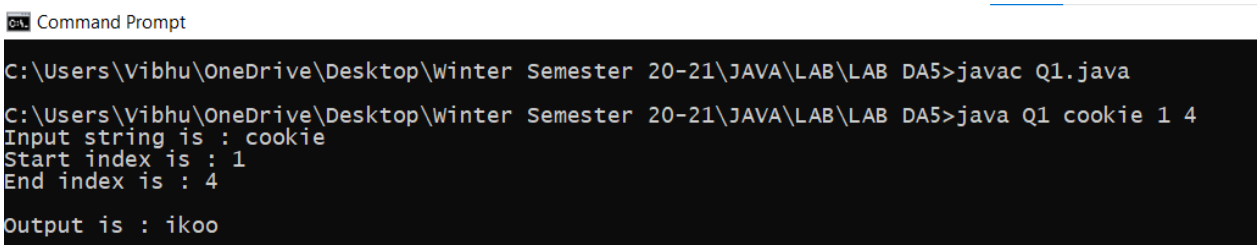
1. **Write a Java program that takes two numbers and string as command line arguments and prints the reverse of the sub-string of the string specified by two numbers. The program should handle all possible exceptions that may arise due to bad inputs. For example, in the command line argument, if a user types Java Substring Cookie 1 4 The output should be 'ikoo'.**

Ans1)

CODE:

```
public class Q1
{
   public static void main(String[] args)
   {
       String input = args[0];
       int start = Integer.parseInt(args[1]);
       int end = Integer.parseInt(args[2]);

       System.out.println("Input string is : "+input);
       System.out.println("Start index is : "+start);
       System.out.println("End index is : "+end);
       if(end<start) {
           System.out.println("Error! Ending index is less than Starting index");
           return ;
       }else if(end>input.length()) {
           System.out.println("Error! Index is out of bound!");
           return ;
       }

       String SubString = input.substring(start,end+1);
       String reverse="";
       for(int i=SubString.length()-1;i>=0;i--) {
           reverse += SubString.charAt(i);
       }

       System.out.println("\nOutput is : "+reverse);
   }
}
```

OUTPUT:



2. **IT Professors are allowed to enter CAT mark and QUIZ mark for students. Professors can enter only CAT mark is between 0 and 100 and Quiz mark is between 0 and 10. Write a Java program that receives the CAT mark and QUIZ mark from Professor. If the marks fail to satisfy the criteria, then handle the exceptions separately for CAT mark and QUIZ mark.**

Ans2)

CODE:

exc.java

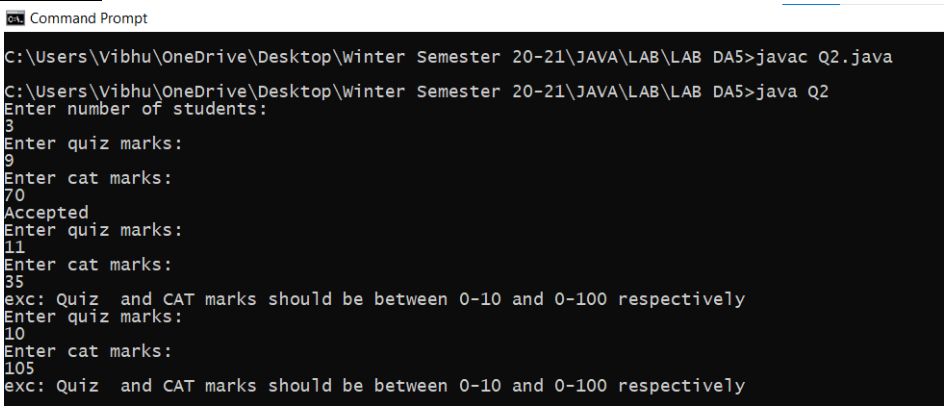
```
public class exc extends Exception
{
    public exc(String s)
    {
        super(s);
    }
}
```

Q2.java

```
import java.util.Scanner;

public class Q2 {
    static Scanner sc =new Scanner(System.in);
    static void marks() {
        System.out.println("Enter number of students: ");
        int n,cat,quiz;
        n =sc.nextInt();

        int[] q = new int[n];
        int[] c = new int[n];
        for(int i=0;i<n;i++){
            try {
                System.out.println("Enter quiz marks:");
                quiz = sc.nextInt();
                System.out.println("Enter cat marks:");
                cat = sc.nextInt();
                if (quiz<0 || quiz>10 || cat<0 || cat > 100) {
                    throw new exc("Quiz  and CAT marks should be between 0-
10 and 0-100 respectively");
                }
                else {
                    System.out.println("Accepted");
                    c[i] = cat;
                    q[i] = quiz;
                }
            } catch (Exception e) {
                System.out.println(e);

            }
        }

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

```
            marks();
        }
    }
```

OUTPUT:



---

3. Write a Java program using threads to compute the first 15 natural numbers, and to compute the first 50 Fibonacci numbers. Set the priority of thread that computes Fibonacci number to 9 and the other to 5. After calculating 30 Fibonacci numbers, make that thread to sleep and take up natural number computation. After computing the 15 natural numbers continue the Fibonacci number computing.

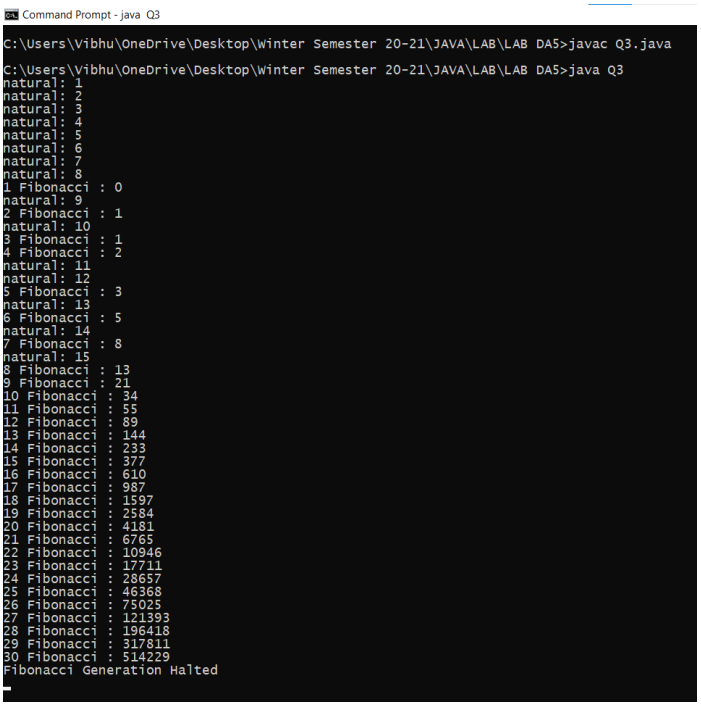
Ans3)

CODE:

```
class Fibonacci{
    private int n=1,a=-1,b=1,c;
    synchronized void disp(){
        for(int i=0;i<=45;i++){
            if(n==31)
                try{
                    System.out.println("Fibonacci Generation Halted");
                    Thread.sleep(5000);
                }catch(InterruptedException e){
                    System.out.println("Caught interrupted exception");
                }
            c=a+b;
            System.out.println(n+" Fibonacci : "+c);
            a=b;
            b=c;
            n++;
        }
    }
}
class Natural{
    int n=1;
    boolean isPrime=true;
    synchronized void disp(){
        for(int i=0;i<15;i++)
        {
            System.out.println("natural: "+ n);
```

```
            n++;
        }
    }
}
class NaturalThread implements Runnable{
    Thread t;
    Natural p1;
    NaturalThread(){
        t=new Thread(this);
        t.setPriority(Thread.NORM_PRIORITY);
        t.start();
    }
    public void run(){
        p1=new Natural();
        p1.disp();
    }
}
class FiboThread implements Runnable{
    Thread t2;
    Fibonacci f;
    FiboThread(){
        t2=new Thread(this);
        t2.setPriority(9);
        t2.start();
    }
    public void run(){
        f=new Fibonacci();
        f.disp();
    }
}
class Q3
{
        public static void main(String args[]){
        FiboThread ft=new FiboThread();
        NaturalThread pt=new NaturalThread();
    }
}
```

OUTPUT:



4. **The southern railway offers concessions for the passengers to travel for a month by giving 25% off on rate for male above 65 years of age and 10% off for female above 60 years of age and 5% off to couples if female is above 18 years and male is above 21years . Create a User defined Exception class so that if the age and gender of the person is not matching with the norms of the concessions, it throws an exception else it offers the concession to the passenger.**

Ans4)

CODE:

CustomException.java

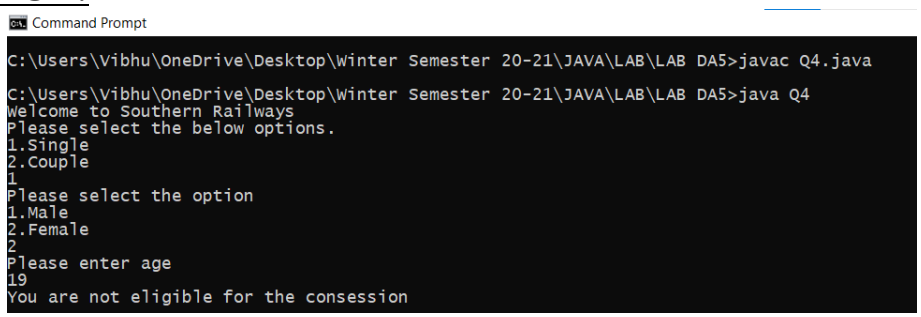
```
public class CustomException extends Exception{
    String str1;
    public CustomException(String str2)
    {
        str1=str2;
    }
    public String toString()
    {
        return (str1) ;
    }
}
```

Q4.java

```
import java.util.Scanner;

public class Q4
{
    public static void main(String args[])
    {
        int options,age,couple1,couple2,gender;
        float discount;
        Scanner sc = new Scanner(System.in);
        try
        {
            System.out.println("Welcome to Southern Railways");
            System.out.println("Please select the below options.");
            System.out.println("1.Single");
            System.out.println("2.Couple");
            options = sc.nextInt();
            if(options==1) //singles
            {
                System.out.println("Please select the option");
                System.out.println("1.Male");
                System.out.println("2.Female");
                gender=sc.nextInt();
                if(gender==1) //male
                {
                    System.out.println("Please enter age");
                    age = sc.nextInt();
                    if(age>65)
                    {
                        discount=25;
```

```
                        System.out.println("You got 25% discount on your ticket")
;
                    }
                    else
                        throw new CustomException("You are not eligible for the c
onsession");
                }
                else
                {
                    System.out.println("Please enter age");
                    age = sc.nextInt();
                    if(age>60)
                    {
                        discount=10;
                        System.out.println("You got 10% discount on your ticket")
;
                    }
                    else
                        throw new CustomException("You are not eligible for the c
onsession");
                }
            }
            else if(options==2) // couples
            {
                System.out.println("Please enter male's age");
                couple1=sc.nextInt();
                System.out.println("Please enter female's age");
                couple2=sc.nextInt();
                if(couple1>21 && couple2>18)
                {
                    discount=5;
                    System.out.println("You both got 5% discount on your ticket")
;
                }
                else
                    throw new CustomException("You are not eligible for the conse
ssion");
            }
            else
            {
                System.out.println("Please enter proper option");
            }
        }
        catch(CustomException exp){
            System.out.println(exp) ;
        }
    }
}
```

OUTPUT:



5. **Mother prepares chapati for her kids. Mother makes chapati and stacks it up in a vessel, and kids eats from it. The max capacity of the vessel is 10. If chapati in the vessel is empty, kids wait for mother to prepare new chapati. Write a Java program to illustrate the given scenario using multithreading.**

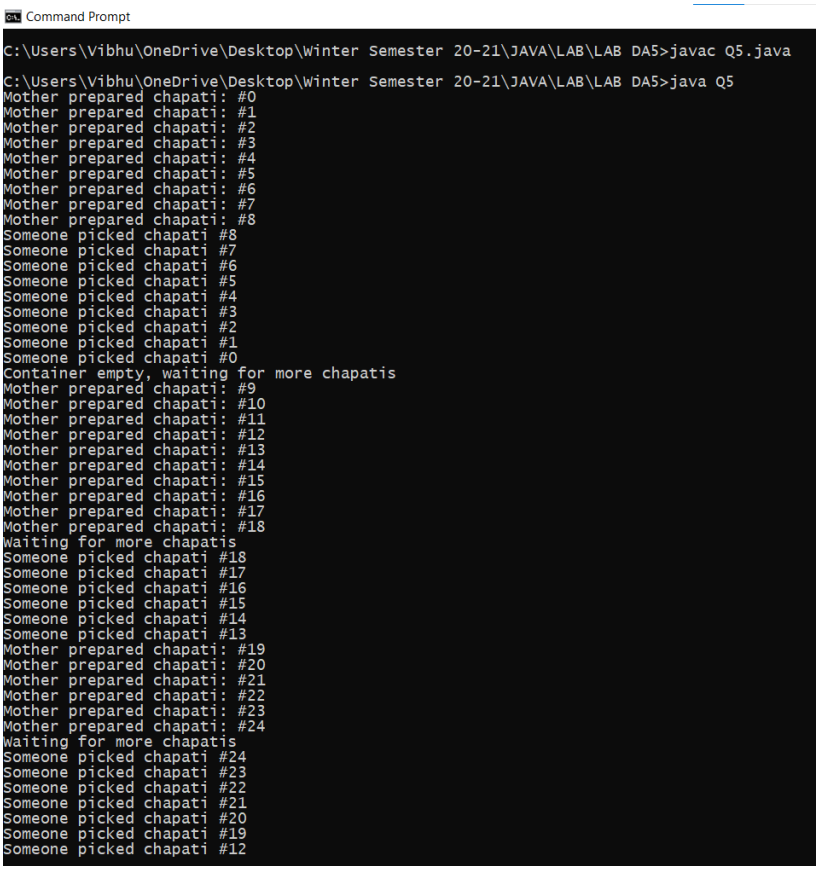
Ans5)

CODE:

```
import java.util.Stack;

class ProducerConsumerWrapper
{
    Stack<Integer> container = new Stack<Integer>();
    int capacity = 10;
    int itemsInStack = 0;
    public void produce() throws InterruptedException
    {

    int value = 0;
    while (true) {
        synchronized (this) {
            while (itemsInStack == capacity)
            {
                System.out.println("Waiting for more chapatis");
            wait();
            }
        System.out.println("Mother prepared chapati: #" + value);
container.push(value++);
itemsInStack++;
notify();
Thread.sleep(1000);
}
}
}
public void consume() throws InterruptedException {
while (true) {
synchronized (this) {
while (container.isEmpty()) {
System.out.println("Container empty, waiting for more chapatis");
wait();
}
int value = container.pop();
itemsInStack--;
System.out.println("Someone picked chapati #" + value);
notify();
Thread.sleep(1000);
}
}
}
}
public class Q5 {
    public static void main(String[] args) throws InterruptedException {
    final ProducerConsumerWrapper a = new ProducerConsumerWrapper();
    Thread producerThread = new Thread(new Runnable() {
    @Override
    public void run() {
```

```
        try {
        a.produce();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        }

        });
        Thread consumerThread = new Thread(new Runnable() {
        @Override
        public void run() {
        try {
        a.consume();
        } catch (InterruptedException e) {
        e.printStackTrace();
        }
        }
        });
        producerThread.start();
        consumerThread.start();
        producerThread.join();
        consumerThread.join();
        }
}
```

OUTPUT:



6. Write a Java program to create a class Student with Registration number, name, CGPA and Proctor Name as its data members. Store the state of objects of this class in a file. Write another class that reads the objects of the Student class from the file. For each object of the class stored in the file, check the CGPA of the student. If the CGPA of a student exceeds 90, then categorize the student grade is "A". If the CGPA of a student is between 70 and 90, then categorize the student grade is "B". If the CGPA of a student is between 50 and 70, then categorize the student grade is "C".

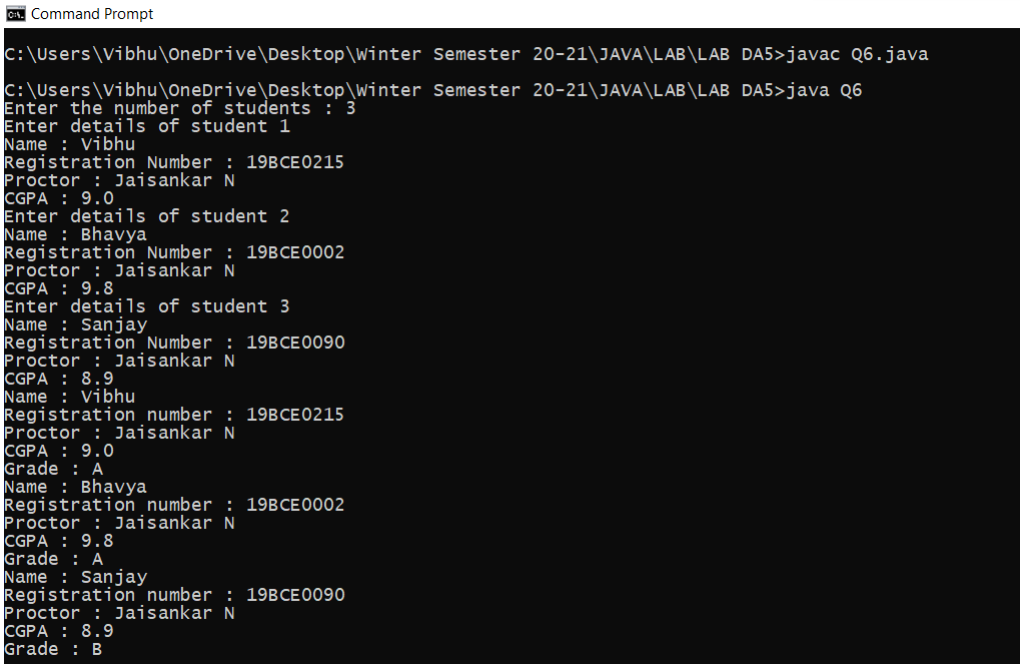
Ans6)

CODE:

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.Scanner;

class Student implements Serializable{
    String regno,name,proctor;
    double cgpa;
    Student(String name,String regno,String proctor,double cgpa)
    {
        this.name=name;
        this.regno=regno;
        this.proctor=proctor;
        this.cgpa=cgpa;
    }

    public void display(){
        System.out.println("Name : "+name);
        System.out.println("Registration number : "+ regno);
        System.out.println("Proctor : "+proctor);
        System.out.println("CGPA : "+cgpa);
        System.out.print("Grade : ");
        if(cgpa>=9)
            System.out.println("A");
        else if(cgpa<9 && cgpa>=7)
            System.out.println("B");
        else if(cgpa<7 && cgpa>5)
            System.out.println("C");
        else
            System.out.println("F");
    }
}

class StudentDisplay implements Serializable{
    StudentDisplay(int n) throws Exception{
        FileInputStream fin=new FileInputStream("student.txt");
        ObjectInputStream in=new ObjectInputStream(fin);
        Student[] s=new Student[n];
        for (int i = 0; i < n; i++) {
            s[i]=(Student) in.readObject();
```

```
            s[i].display();
        }
    }

}
public class Q6{
    public static void main(String[] args) throws Exception {
        System.out.print("Enter the number of students : ");
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        sc.nextLine();
        FileOutputStream fout = new FileOutputStream("student.txt");
        ObjectOutputStream out = new ObjectOutputStream(fout);
        Student[] s=new Student[n];
        for (int i = 0; i < n; i++) {
            System.out.println("Enter details of student "+(i+1));
            System.out.print("Name : ");
            String name=sc.nextLine();
            System.out.print("Registration Number : ");
            String regno=sc.nextLine();
            System.out.print("Proctor : ");
            String proctor=sc.nextLine();
            System.out.print("CGPA : ");
            double cgpa=sc.nextDouble();
            sc.nextLine();
            s[i]=new Student(name,regno,proctor,cgpa);
            out.writeObject(s[i]);
        }
        new StudentDisplay(n);
    }
}
```

OUTPUT:



7. Write a Java program to define a class 'CovaxinVaccineCamp to store the below mentioned details for those who were vaccinated at VIT health center. Emp-id, Name, age, address, mobile number, blood group, Allergy (yes/no), Date of faculty vaccinated, Create 'n' objects of this class for all the vaccinated. faculty at VIT, Vellore. Write these objects into a file. Read these objects from the file and display the faculty details those who are having allergy.

Ans 7)

CODE:

```
import java.io.*;
import java.util.*;

class Faculty implements Serializable {
    String emp;
    String name;
    int age;
    String address;
    long contact;
    String allergy;
    String date;

    Faculty(String emp, String name, int age, String address, long contact, Strin
g allergy, String date2) {
        this.emp = emp;
        this.name = name;
        this.age = age;
        this.address = address;
        this.contact = contact;
        this.allergy = allergy;
        this.date = date2;
    }

    void display() {
        System.out.println("Employee id: " + emp);
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Address: " + address);
        System.out.println("Contact Number: " + contact);
        System.out.println("Allergy: " + allergy);
        System.out.println("Date of Vaccination" + date);
        System.out.println();
    }
}

public class Q7{
    public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(System.in);
        int n;
        System.out.print("Enter no. of faculties: ");
        n = sc.nextInt();
        Faculty[] f = new Faculty[n];
        Faculty[] faculty_new = new Faculty[n];
        FileOutputStream fout = new FileOutputStream("lol.txt");
        ObjectOutputStream oout = new ObjectOutputStream(fout);
```

```
            FileInputStream fin = new FileInputStream("lol.txt");
            ObjectInputStream oin = new ObjectInputStream(fin);
            for (int i = 0; i < n; i++) {
                System.out.println("Details of faculty " + (i + 1));
                System.out.print("Enter employee id: ");
                String Empid = sc.next();
                System.out.print("Enter Name: ");
                String Name = sc.next();
                System.out.print("Enter Age: ");
                int Age = sc.nextInt();
                System.out.print("Enter Address: ");
                String Address = sc.next();
                System.out.print("Enter Contact number: ");
                long Contact = sc.nextLong();
                System.out.print("Enter yes if Allergy: ");
                String Allergy = sc.next();
                System.out.print("Enter Date of vaccination: ");
                String date = sc.next();
                f[i] = new Faculty(Empid, Name, Age, Address, Contact, Allergy, date)
;
                oout.writeObject(f[i]);
                System.out.println();
            }
            System.out.println("Details of faculty with allergy: ");
            for (int i = 0; i < n; i++) {
                faculty_new[i] = (Faculty) oin.readObject();
                if (faculty_new[i].allergy.equals("yes")) {
                    f[i].display();
                }
            }
            sc.close();
            oout.close();
            fin.close();
            System.out.println("Success!");
        }
}
```

OUTPUT:

Command Prompt

```
C:\Users\Vibhu\OneDrive\Desktop\Winter Semester 20-21\JAVA\LAB\LAB DA5>javac Q7.java

C:\Users\Vibhu\OneDrive\Desktop\Winter Semester 20-21\JAVA\LAB\LAB DA5>java Q7
Enter no. of faculties: 2
Details of faculty 1
Enter employee id: 001
Enter Name: Jaisankar
Enter Age: 30
Enter Address: Vellore
Enter Contact number: 9090909090
Enter yes if Allergy: yes
Enter Date of vaccination: 21-05-2021

Details of faculty 2
Enter employee id: 002
Enter Name: SanthiH
Enter Age: 50
Enter Address: Vellore
Enter Contact number: 9191919191
Enter yes if Allergy: yes
Enter Date of vaccination: 21-04-2021

Details of faculty with allergy:
Employee id: 001
Name: Jaisankar
Age: 30
Address: Vellore
Contact Number: 9090909090
Allergy: yes
Date of Vaccination21-05-2021

Employee id: 002
Name: SanthiH
Age: 50
Address: Vellore
Contact Number: 9191919191
Allergy: yes
Date of Vaccination21-04-2021

Success!
```

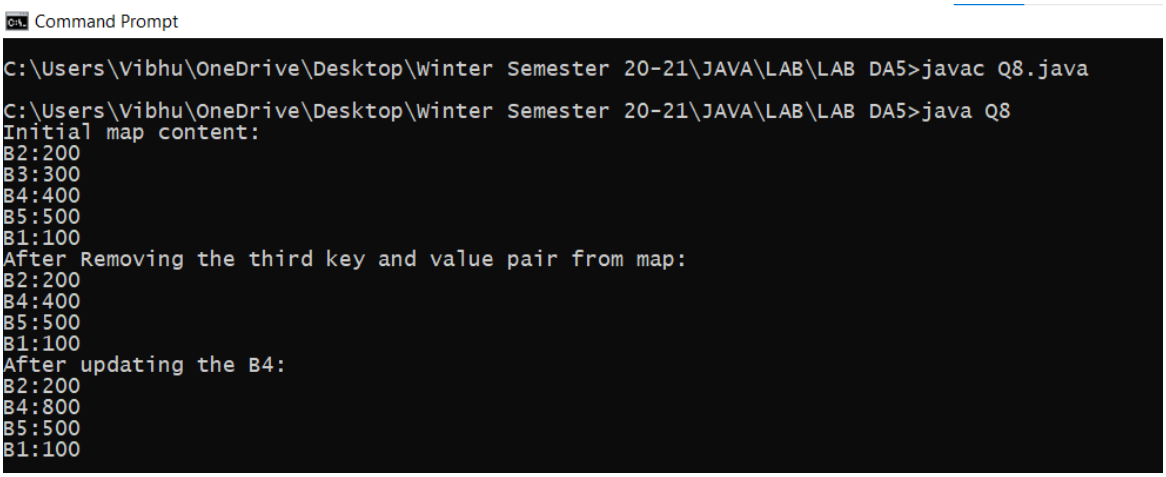
8. **Create a map to store book-Id and cost with five sample values with keys associated with it, like (B1, 100), (B2, 200), (B3, 300), (B4, 400) and (B5, 500). Remove the third Key and Value pair from the Map, also update the cost of B4 value as 800. Traverse through the Map elements using lambda expression.**

Ans8)

CODE:

```
import java.util.*;
public class Q8 {
    public static void main(String[] args){
        Map<String,Integer> map=new HashMap<>();
        map.put("B1",100);
        map.put("B2",200);
        map.put("B3",300);
        map.put("B4",400);
        map.put("B5",500);
        System.out.println("Initial map content:");
        map.forEach((k, v) -> System.out.println((k + ":" + v)));
        System.out.println("After Removing the third key and value pair from map:
");
        map.remove("B3");
        map.forEach((k, v) -> System.out.println((k + ":" + v)));
        System.out.println("After updating the B4:");
        map.replace("B4",800);
        map.forEach((k, v) -> System.out.println((k + ":" + v)));
    }
}
```

OUTPUT:



---

9. **(i) Write a Java program using generic method reverseArray that reverses the order of elements in an array of different types. Print each array before and after calling reverseArray method.(6)**

(ii) Write a Java program to read book id, author and publisher details and add books to list and printing all the books using ArrayList and Iterator.

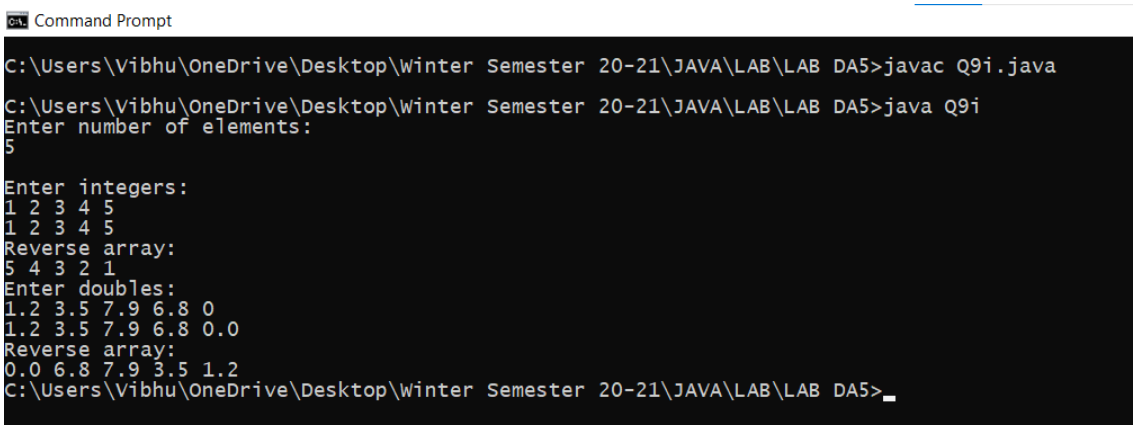
Ans9)

(i)

CODE:

```
import java.util.Scanner;

public class Q9i {
    static Scanner sc = new Scanner(System.in);

    static < E > void ReverseArray( E[] inputArray ) {
        int len =inputArray.length;
        System.out.println("\nReverse array:");
        for(int i=len-1;i>=0;i--) {
            System.out.printf("%s ", inputArray[i]);
        }
    }
    public static void main(String[] args) {
        System.out.println("Enter number of elements:");
        int n;
        n =sc.nextInt();
        Integer[] num = new Integer[n];
        Double[] dou = new Double[n];

        System.out.println("\nEnter integers: ");
        for(int i=0;i<n;i++)
        {
            num[i] = sc.nextInt();
        }

        for(int i=0;i<n;i++)
        {
            System.out.print(num[i]+" ");
        }
        ReverseArray(num);
        System.out.println("\nEnter doubles: ");
        for(int i=0;i<n;i++) {
            dou[i] = sc.nextDouble();
        }
        for(int i=0;i<n;i++)
        {
            System.out.print(dou[i]+" ");
        }
        ReverseArray(dou);
    }
}
```

OUTPUT:

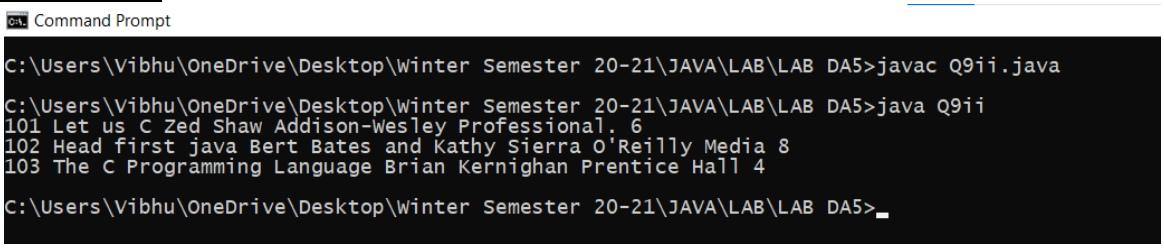


(ii)

CODE:

```
import java.util.*;
class Book {
int id;
String name,author,publisher;
int quantity;
public Book(int id, String name, String author, String publisher, int quantity) {
    this.id = id;
    this.name = name;
    this.author = author;
    this.publisher = publisher;
    this.quantity = quantity;
}
}
public class Q9ii {
public static void main(String[] args) {
    //Creating list of Books
    List<Book> list=new ArrayList<Book>();
    Book b1=new Book(101,"Let us C","Zed Shaw","Addison-Wesley Professional.",6);
    Book b2=new Book(102,"Head first java","Bert Bates and Kathy Sierra","O'Reilly M
edia",8);
    Book b3=new Book(103,"The C Programming Language","Brian Kernighan","Prentice Ha
ll",4);
    list.add(b1);
    list.add(b2);
    list.add(b3);
    for(Book b:list){
        System.out.println(b.id+" "+b.name+" "+b.author+" "+b.publisher+" "+b.quanti
ty);
    }
}
}
```

OUTPUT:



10. Simple calculator using JavaFx.

Ans10)

CODE:

```
import javafx.application.Application;
import javafx.beans.binding.Bindings;
import javafx.beans.property.*;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.input.KeyEvent;
import javafx.scene.layout.*;
import javafx.stage.Stage;
import javafx.stage.StageStyle;

import java.util.HashMap;
import java.util.Map;

// a simple JavaFX calculator.
public class Calc extends Application {
    private static final String[][] template = {
            { "9", "8", "9", "/" },
            { "1", "5", "6", "*" },
            { "1", "2", "3", "-" },
            { "0", "c", "=", "+" }
    };

    private final Map<String, Button> accelerators = new HashMap<>();

    private DoubleProperty stackValue = new SimpleDoubleProperty();
    private DoubleProperty value = new SimpleDoubleProperty();

    private enum Op { NOOP, ADD, SUBTRACT, MULTIPLY, DIVIDE }

    private Op curOp   = Op.NOOP;
    private Op stackOp = Op.NOOP;

    public static void main(String[] args) { launch(args); }

    @Override public void start(Stage stage) {
        final TextField screen  = createScreen();
        final TilePane  buttons = createButtons();

        stage.setTitle("Calc");
        stage.initStyle(StageStyle.UTILITY);
        stage.setResizable(false);
        stage.setScene(new Scene(createLayout(screen, buttons)));
        stage.show();
    }

    private VBox createLayout(TextField screen, TilePane buttons) {
        final VBox layout = new VBox(20);
        layout.setAlignment(Pos.CENTER);
        layout.setStyle("-fx-background-color: chocolate; -fx-padding: 20; -fx-font-
size: 20;");
        layout.getChildren().setAll(screen, buttons);
```

```
        handleAccelerators(layout);
        screen.prefWidthProperty().bind(buttons.widthProperty());
        return layout;
    }

    private void handleAccelerators(VBox layout) {
        layout.addEventFilter(KeyEvent.KEY_PRESSED, new EventHandler<KeyEvent>() {
            @Override
            public void handle(KeyEvent keyEvent) {
                Button activated = accelerators.get(keyEvent.getText());
                if (activated != null) {
                    activated.fire();
                }
            }
        });
    }

    private TextField createScreen() {
        final TextField screen = new TextField();
        screen.setStyle("-fx-background-color: aquamarine;");
        screen.setAlignment(Pos.CENTER_RIGHT);
        screen.setEditable(false);
        screen.textProperty().bind(Bindings.format("%.0f", value));
        return screen;
    }

    private TilePane createButtons() {
        TilePane buttons = new TilePane();
        buttons.setVgap(7);
        buttons.setHgap(7);
        buttons.setPrefColumns(template[0].length);
        for (String[] r: template) {
            for (String s: r) {
                buttons.getChildren().add(createButton(s));
            }
        }
        return buttons;
    }

    private Button createButton(final String s) {
        Button button = makeStandardButton(s);

        if (s.matches("[0-9]")) {
            makeNumericButton(s, button);
        } else {
            final ObjectProperty<Op> triggerOp = determineOperand(s);
            if (triggerOp.get() != Op.NOOP) {
                makeOperandButton(button, triggerOp);
            } else if ("c".equals(s)) {
                makeClearButton(button);
            } else if ("=".equals(s)) {
                makeEqualsButton(button);
            }
        }

        return button;
    }

    private ObjectProperty<Op> determineOperand(String s) {
        final ObjectProperty<Op> triggerOp = new SimpleObjectProperty<>(Op.NOOP);
```

```
            switch (s) {
                case "+": triggerOp.set(Op.ADD);      break;
                case "-": triggerOp.set(Op.SUBTRACT); break;
                case "*": triggerOp.set(Op.MULTIPLY); break;
                case "/": triggerOp.set(Op.DIVIDE);   break;
            }
            return triggerOp;
        }

        private void makeOperandButton(Button button, final ObjectProperty<Op> triggerOp
    ) {
            button.setStyle("-fx-base: lightgray;");
            button.setOnAction(new EventHandler<ActionEvent>() {
                @Override
                public void handle(ActionEvent actionEvent) {
                    curOp = triggerOp.get();
                }
            });
        }

        private Button makeStandardButton(String s) {
            Button button = new Button(s);
            button.setStyle("-fx-base: beige;");
            accelerators.put(s, button);
            button.setMaxSize(Double.MAX_VALUE, Double.MAX_VALUE);
            return button;
        }

        private void makeNumericButton(final String s, Button button) {
            button.setOnAction(new EventHandler<ActionEvent>() {
                @Override
                public void handle(ActionEvent actionEvent) {
                    if (curOp == Op.NOOP) {
                        value.set(value.get() * 10 + Integer.parseInt(s));
                    } else {
                        stackValue.set(value.get());
                        value.set(Integer.parseInt(s));
                        stackOp = curOp;
                        curOp = Op.NOOP;
                    }
                }
            });
        }

        private void makeClearButton(Button button) {
            button.setStyle("-fx-base: mistyrose;");
            button.setOnAction(new EventHandler<ActionEvent>() {
                @Override
                public void handle(ActionEvent actionEvent) {
                    value.set(0);
                }
            });
        }

        private void makeEqualsButton(Button button) {
            button.setStyle("-fx-base: ghostwhite;");
            button.setOnAction(new EventHandler<ActionEvent>() {
                @Override
                public void handle(ActionEvent actionEvent) {
                    switch (stackOp) {
```

```
                    case ADD:      value.set(stackValue.get() + value.get()); break;
                    case SUBTRACT: value.set(stackValue.get() - value.get()); break;
                    case MULTIPLY: value.set(stackValue.get() * value.get()); break;
                    case DIVIDE:   value.set(stackValue.get() / value.get()); break;
                }
            }
        });
    }
}
```

OUTPUT:

Addition:

Multiplication:

---

11. Create menu(menubar, menu and menuitems) and event handler using JavaFx.

Ans11)

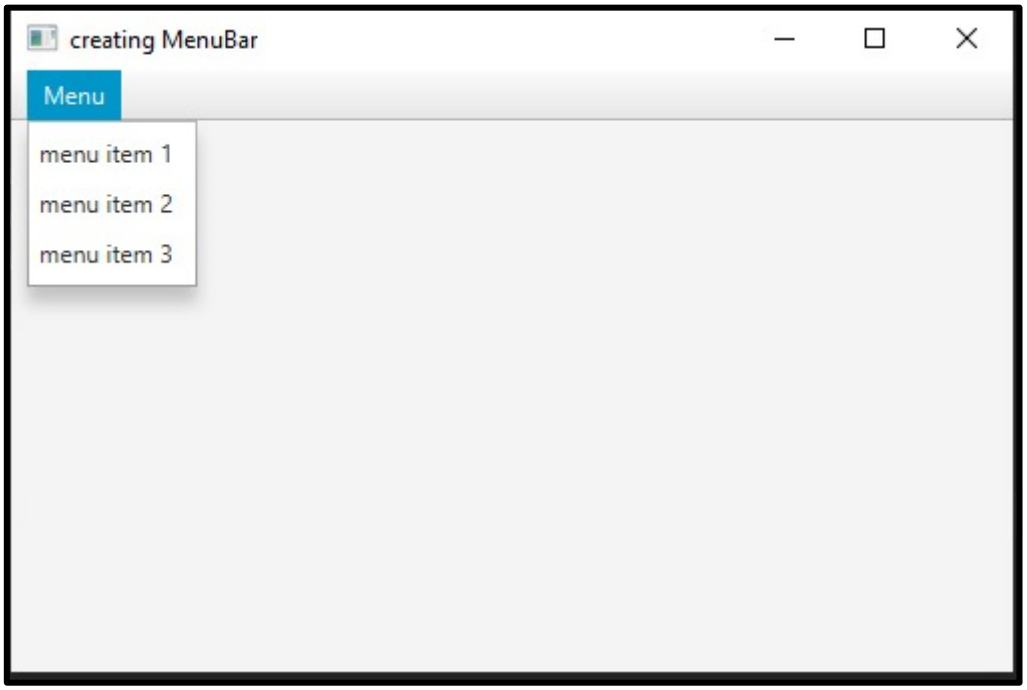
CODE:

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.*;
import javafx.event.ActionEvent;
```

```
import javafx.event.EventHandler;
import javafx.scene.control.*;
import javafx.stage.Stage;
import javafx.scene.control.Alert.AlertType;
import java.time.LocalDate;
public class Da5Ques11 extends Application {

    // launch the application
    public void start(Stage s)
    {
        // set title for the stage
        s.setTitle("creating MenuBar");

        // create a menu
        Menu m = new Menu("Menu");

        // create menuitems
        MenuItem m1 = new MenuItem("menu item 1");
        MenuItem m2 = new MenuItem("menu item 2");
        MenuItem m3 = new MenuItem("menu item 3");

        // add menu items to menu
        m.getItems().add(m1);
        m.getItems().add(m2);
        m.getItems().add(m3);

        // create a menubar
        MenuBar mb = new MenuBar();

        // add menu to menubar
        mb.getMenus().add(m);

        // create a VBox
        VBox vb = new VBox(mb);

        // create a scene
        Scene sc = new Scene(vb, 500, 300);

        // set the scene
        s.setScene(sc);

        s.show();
    }

    public static void main(String args[])
    {
        // launch the application
        launch(args);
    }
}
```

OUTPUT:


creating MenuBar
Menu


creating MenuBar
Menu
menu item 1
menu item 2
menu item 3
