JAVA PROGRAMMING
CSE-1007

LAB DIGITAL ASSIGNMENT 5

Name: Mathew Jerry Meleth

Reg No: 17BIT0050

Slot: L19 + L20

QUESTION 1 :

a.Write a program sort the array using generic function (input int,string ,float)

b.write a program to do stack operation using generic class (input int,string, float)

BRIEF ABOUT YOUR APPROACH:

a)Declare a comparable object

Insert elements into T

Using CompareTo() sort the array(same for all datatypes

SOURCE CODE:

A)

```
import java.util.*;
class genArrSrt<T>
{
   T t;
}
public class GenFuc {
  public static void main(String args[])
   {
      Scanner S = new Scanner(System.in);
      int c=0;
      while(c!=4)
      {
                System.out.print("Choose your
option:\n1.IntergerArraySort\n2.StringArraySort\n3.FloatArraySort\n4.Exit\n");
      c = S.nextInt();
      if(c==1)
      {
                        System.out.println("Enter no of elements in the array: ");
                        int n = S.nextInt();
         genArrSrt<Integer> arr[] = new genArrSrt[n];
         for(int i=0;i<n;i++)
         {
            arr[i] = new genArrSrt();
         }
         System.out.println("Enter the values: ");
         for(int i=0;i<n;i++)
         {
            arr[i].t = S.nextInt();
         }
         for(int i=1;i<n;i++)
         {
            int j=i-1;
            int temp = arr[i].t;
            while(j>=0 && arr[j].t>temp)
            {
               arr[j+1].t = arr[j].t;
               j--;
            }
            arr[j+1].t = temp;
         }
         System.out.print("The Sorted Array is ");
         for(int i=0;i<n;i++)
         {
            System.out.print(arr[i].t+" ");
         }
         System.out.println();
      }
      else if(c==2)
      {
                        System.out.println("Enter no of elements in the array: ");
                        int n = S.nextInt();
         genArrSrt<String> arr[] = new genArrSrt[n];
         for(int i=0;i<n;i++)
         {
            arr[i] = new genArrSrt();
         }
         System.out.println("Enter the values: ");
         for(int i=0;i<n;i++)
         {
            arr[i].t = S.next();
         }
         for(int i=1;i<n;i++)
         {
            int j=i-1;
            String temp = arr[i].t;
            while(j>=0 && arr[j].t.compareTo(temp)>0)
            {
               arr[j+1].t = arr[j].t;
               j--;
            }
            arr[j+1].t = temp;
         }
         System.out.print("The Sorted Array is ");
         for(int i=0;i<n;i++)
         {
            System.out.print(arr[i].t+" ");
         }
         System.out.println();
      }
      else if(c==3)
      {
                        System.out.println("Enter no of elements in the array: ");
                        int n = S.nextInt();
         genArrSrt<Float> arr[] = new genArrSrt[n];
         for(int i=0;i<n;i++)
         {
            arr[i] = new genArrSrt();
```

```
         }
         System.out.println("Enter the values: ");
         for(int i=0;i<n;i++)
         {
            arr[i].t = S.nextFloat();
         }
         for(int i=1;i<n;i++)
         {
            int j=i-1;
            float temp = arr[i].t;
            while(j>=0 && arr[j].t>temp)
            {
               arr[j+1].t = arr[j].t;
               j--;
            }
            arr[j+1].t = temp;
         }
         System.out.print("The Sorted Array is ");
         for(int i=0;i<n;i++)
         {
            System.out.print(arr[i].t+" ");
         }
         System.out.println();
      }
                 }
   }
}
```

===

B)

```
import java.util.*;
class stackOperations<T>
{
   T t;
}
public class GenFuc2
{
   public static void main(String args[])
   {
      Scanner S = new Scanner(System.in);
      int top = -1;
      System.out.print("Choose your
option:\n1.IntergerStackOperations\n2.StringStackOperations\n3.FloatStackOperation
s\n");
      int choice = S.nextInt();
```

```
        System.out.println("Enter the size of the stack: ");
        int n = S.nextInt();
        if(choice==1)
        {
                                stackOperations<Integer> arr[] = new stackOperations[n];
                                for(int i=0;i<n;i++)
                                {
                                           arr[i] = new stackOperations();
                                }
                                int c=0;
        while(c!=4)
        {
            System.out.println("Choose your option for stack
operation:\n1.Push\n2.Pop\n3.Display\n4.Exit");
            c = S.nextInt();
            switch(c)
            {
                case 1:
                {
                    System.out.print("Enter the value to be pushed: ");
                    int val = S.nextInt();
                    System.out.println();
                    if(top == n-1)
                    {
                        System.out.println("Stack is full");
                    }
                    else
                    {
                        arr[++top].t = val;
                        System.out.println(val+" is pushed into stack");
                    }
                    break;
                }
                case 2:
                {
                    if(top==-1)
                    {
                        System.out.println("Stack is empty");
                    }
                    else
                    {
                        System.out.println(arr[top--].t+" is popped out of the stack");
                    }
                    break;
                }
                case 3:
```

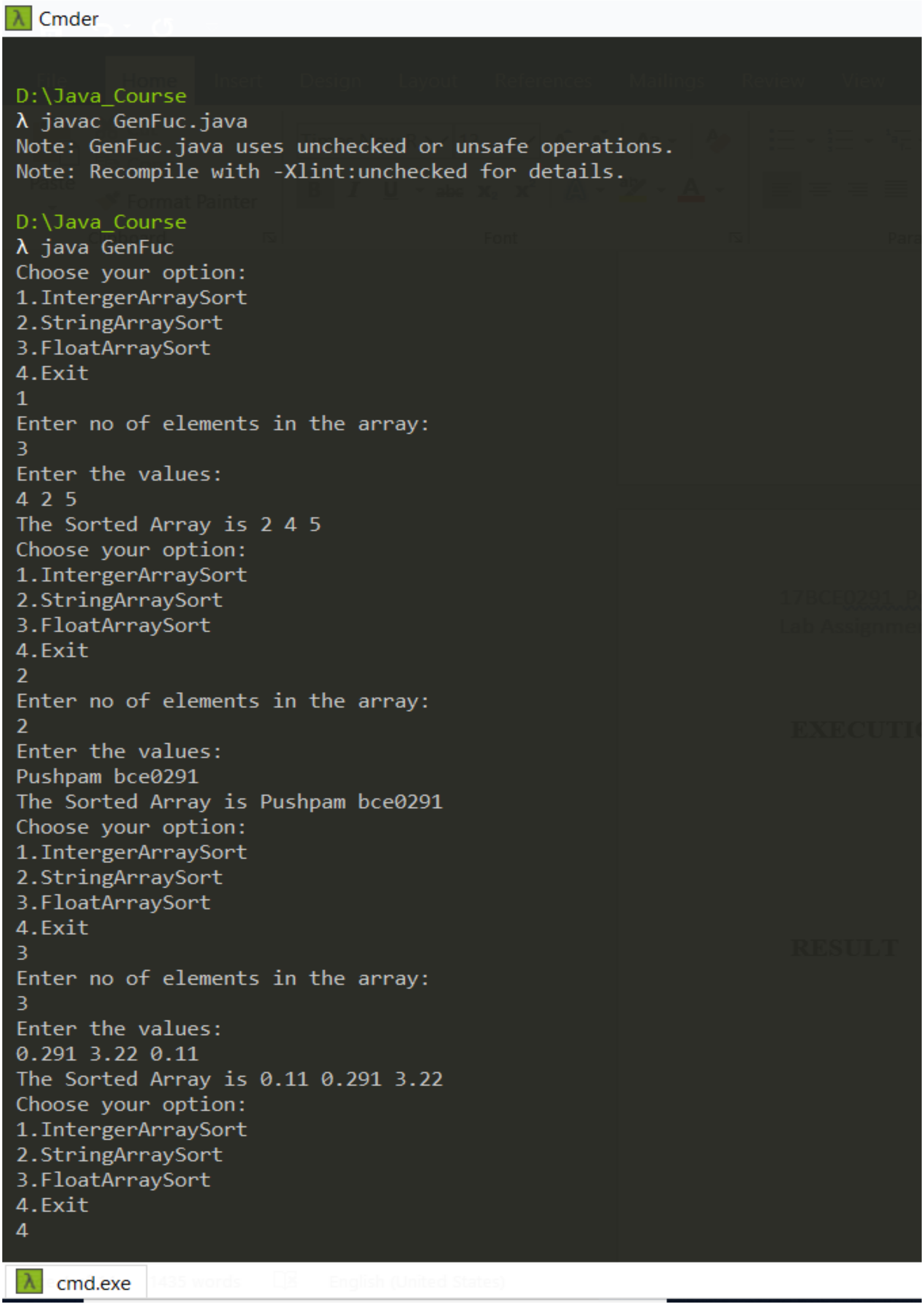
```
        {
          if(top==-1)
          {
            System.out.println("Stack is empty");
          }
          else
          {
            System.out.println("Stack elements are ");
            for(int i=top;i>=0;i--)
            {
              System.out.print(arr[i].t+" ");
            }
            System.out.println();
          }
          break;
        }
      }
    }
           }
           else if(choice==2)
           {
                 stackOperations<String> arr[] = new stackOperations[n];
                 for(int i=0;i<n;i++)
                 {
                      arr[i] = new stackOperations();
                 }
                 int c=0;
    while(c!=4)
    {
      System.out.println("Choose your option for stack
operation:\n1.Push\n2.Pop\n3.Display\n4.Exit");
      c = S.nextInt();
      switch(c)
      {
        case 1:
        {
          System.out.print("Enter the value to be pushed: ");
          String val = S.next();
          System.out.println();
          if(top == n-1)
          {
            System.out.println("Stack is full");
          }
          else
          {
            arr[++top].t = val;
            System.out.println(val+" is pushed into stack");
         }
         break;
      }
      case 2:
      {
         if(top==-1)
         {
            System.out.println("Stack is empty");
         }
         else
         {
            System.out.println(arr[top--].t+" is popped out of the stack");
         }
         break;
      }
      case 3:
      {
         if(top==-1)
         {
            System.out.println("Stack is empty");
         }
         else
         {
            System.out.println("Stack elements are ");
            for(int i=top;i>=0;i--)
            {
               System.out.print(arr[i].t+" ");
            }
            System.out.println();
         }
         break;
      }
   }
}
          }
          else if(choice==3)
          {
                  stackOperations<Float> arr[] = new stackOperations[n];
                  for(int i=0;i<n;i++)
                  {
                           arr[i] = new stackOperations();
                  }
                  int c=0;
while(c!=4)
{
```

```
            System.out.println("Choose your option for stack 
operation:\n1.Push\n2.Pop\n3.Display\n4.Exit");
            c = S.nextInt();
            switch(c)
            {
               case 1:
               {
                  System.out.print("Enter the value to be pushed: ");
                  float val = S.nextFloat();
                  System.out.println();
                  if(top == n-1)
                  {
                     System.out.println("Stack is full");
                  }
                  else
                  {
                     arr[++top].t = val;
                     System.out.println(val+" is pushed into stack");
                  }
                  break;
               }
               case 2:
               {
                  if(top==-1)
                  {
                     System.out.println("Stack is empty");
                  }
                  else
                  {
                     System.out.println(arr[top--].t+" is popped out of the stack");
                  }
                  break;
               }
               case 3:
               {
                  if(top==-1)
                  {
                     System.out.println("Stack is empty");
                  }
                  else
                  {
                     System.out.println("Stack elements are ");
                     for(int i=top;i>=0;i--)
                     {
                        System.out.print(arr[i].t+" ");
                     }
```

```
                        System.out.println();
                    }
                    break;
                }
            }
        }
                      }
    }
}
```

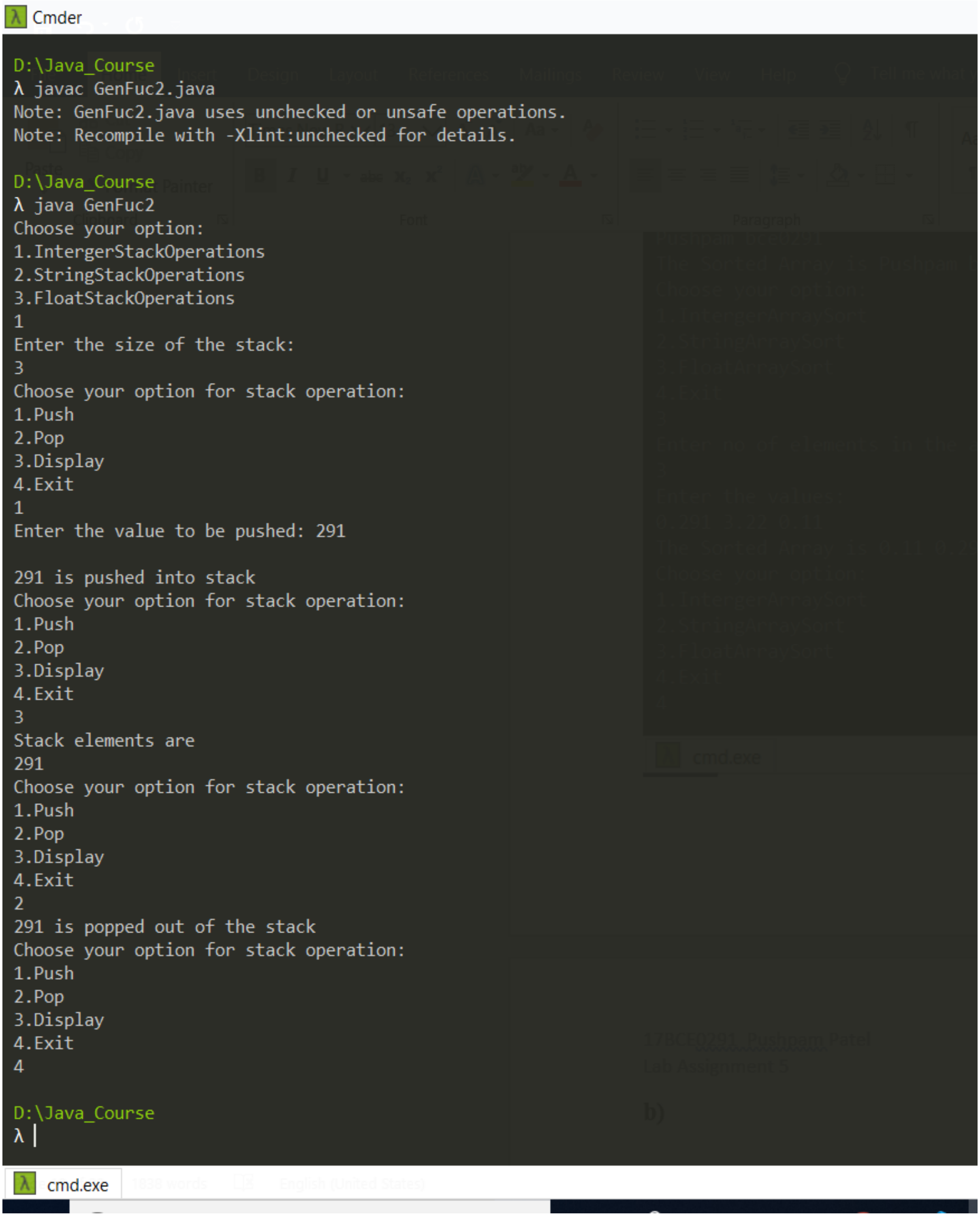
EXECUTION:

a)



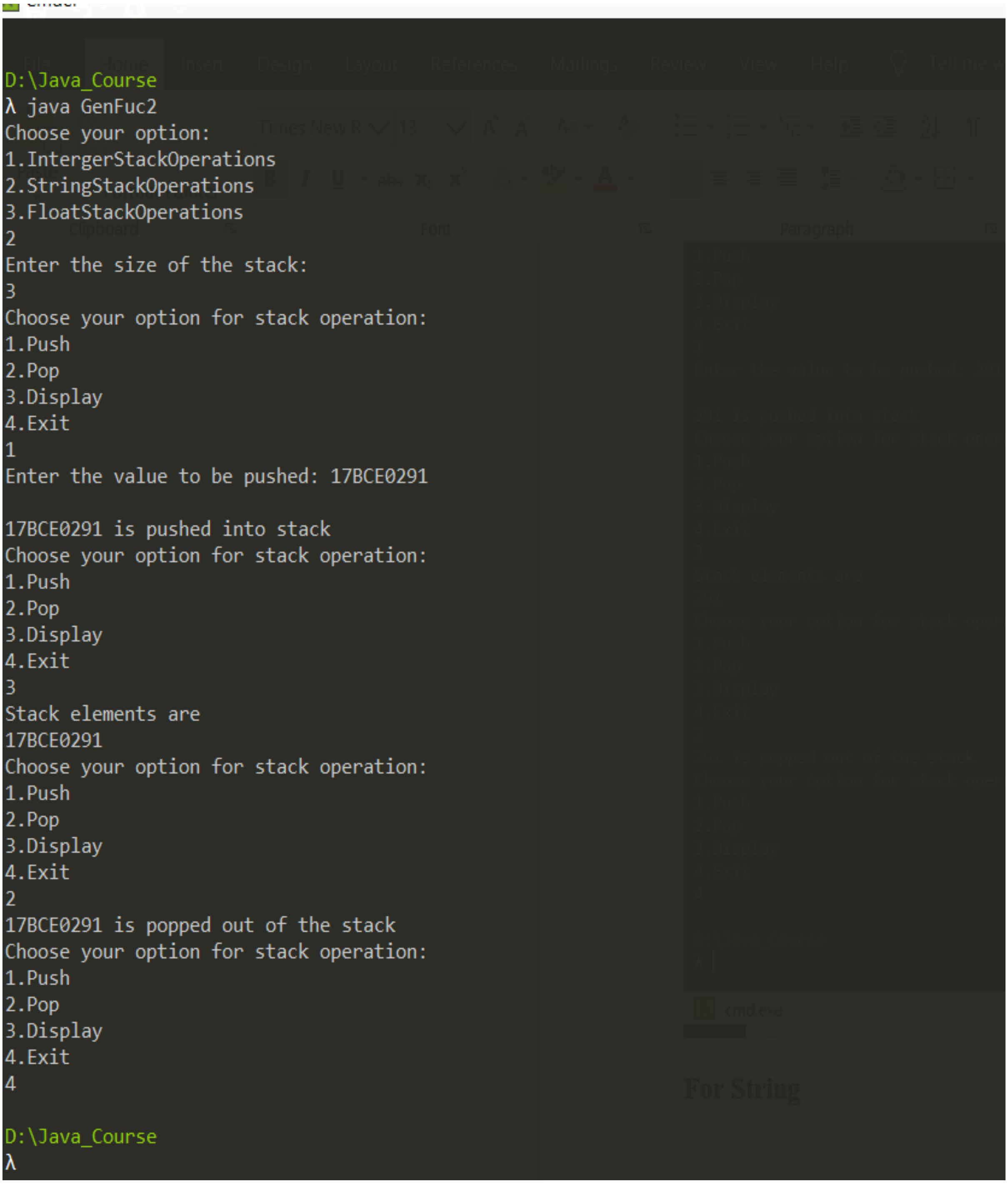
```
D:\Java_Course
λ javac GenFuc.java
Note: GenFuc.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

D:\Java_Course
λ java GenFuc
Choose your option:
1.IntergerArraySort
2.StringArraySort
3.FloatArraySort
4.Exit
1
Enter no of elements in the array:
3
Enter the values:
4 2 5
The Sorted Array is 2 4 5
Choose your option:
1.IntergerArraySort
2.StringArraySort
3.FloatArraySort
4.Exit
2
Enter no of elements in the array:
2
Enter the values:
Pushpam bce0291
The Sorted Array is Pushpam bce0291
Choose your option:
1.IntergerArraySort
2.StringArraySort
3.FloatArraySort
4.Exit
3
Enter no of elements in the array:
3
Enter the values:
0.291 3.22 0.11
The Sorted Array is 0.11 0.291 3.22
Choose your option:
1.IntergerArraySort
2.StringArraySort
3.FloatArraySort
4.Exit
4
```

b)
For Integer



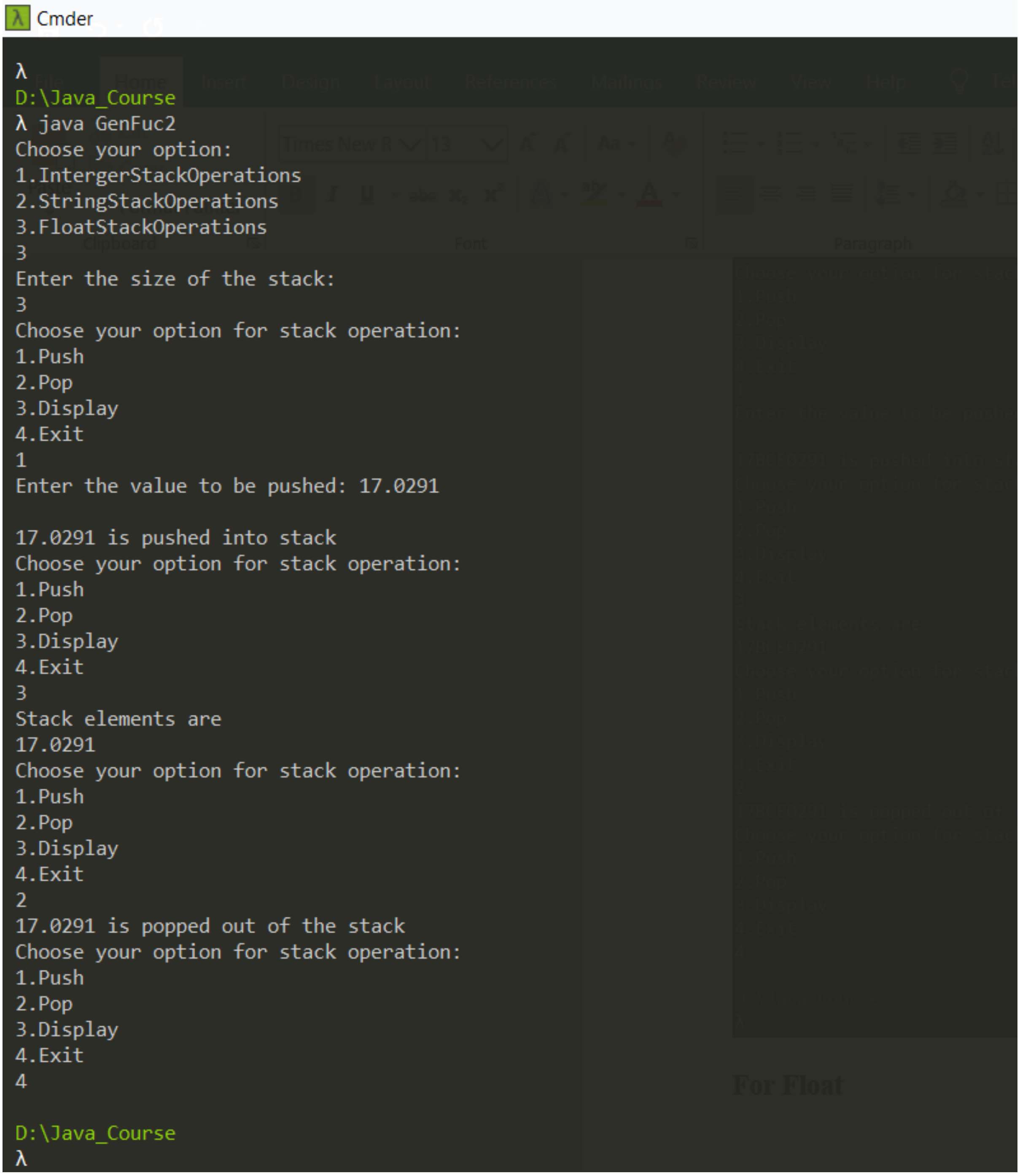
```
D:\Java_Course
λ javac GenFuc2.java
Note: GenFuc2.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

D:\Java_Course
λ java GenFuc2
Choose your option:
1.IntergerStackOperations
2.StringStackOperations
3.FloatStackOperations
1
Enter the size of the stack:
3
Choose your option for stack operation:
1.Push
2.Pop
3.Display
4.Exit
1
Enter the value to be pushed: 291

291 is pushed into stack
Choose your option for stack operation:
1.Push
2.Pop
3.Display
4.Exit
3
Stack elements are
291
Choose your option for stack operation:
1.Push
2.Pop
3.Display
4.Exit
2
291 is popped out of the stack
Choose your option for stack operation:
1.Push
2.Pop
3.Display
4.Exit
4

D:\Java_Course
λ
```

For String



```
D:\Java_Course
λ java GenFuc2
Choose your option:
1.IntergerStackOperations
2.StringStackOperations
3.FloatStackOperations
2
Enter the size of the stack:
3
Choose your option for stack operation:
1.Push
2.Pop
3.Display
4.Exit
1
Enter the value to be pushed: 17BCE0291

17BCE0291 is pushed into stack
Choose your option for stack operation:
1.Push
2.Pop
3.Display
4.Exit
3
Stack elements are
17BCE0291
Choose your option for stack operation:
1.Push
2.Pop
3.Display
4.Exit
2
17BCE0291 is popped out of the stack
Choose your option for stack operation:
1.Push
2.Pop
3.Display
4.Exit
4

D:\Java_Course
λ
```

For Float



```
λ
D:\Java_Course
λ java GenFuc2
Choose your option:
1.IntergerStackOperations
2.StringStackOperations
3.FloatStackOperations
3
Enter the size of the stack:
3
Choose your option for stack operation:
1.Push
2.Pop
3.Display
4.Exit
1
Enter the value to be pushed: 17.0291

17.0291 is pushed into stack
Choose your option for stack operation:
1.Push
2.Pop
3.Display
4.Exit
3
Stack elements are
17.0291
Choose your option for stack operation:
1.Push
2.Pop
3.Display
4.Exit
2
17.0291 is popped out of the stack
Choose your option for stack operation:
1.Push
2.Pop
3.Display
4.Exit
4

D:\Java_Course
λ
```

RESULT:

a)

Sorted array of each format is been printed, after entering the size and the elements of the array

b)

For all the format all the operations are been working properly as shown in the above Output screen

QUESTION 2 :

Write a java program to create class **Books** with the data members **book name, author, price, type(fiction, comic, cooking)**. Use input methods to get the input values. Create three array list (fiction, comic and cooking). [10M]

a. Depending upon the type of the book, insert the book object into the respective list.
b. Display the list of books in each type.
c. Sort the list of books in each list with respect to their book name.
d. Display the min and max priced books of each list.

BRIEF ABOUT YOUR APPROACH:

Create a 3 arraylists with type as a data member.

Using an if condition check for the type and insert into the respective list.

Using CompareTo sort the list.

Using Comparator find the max and min.

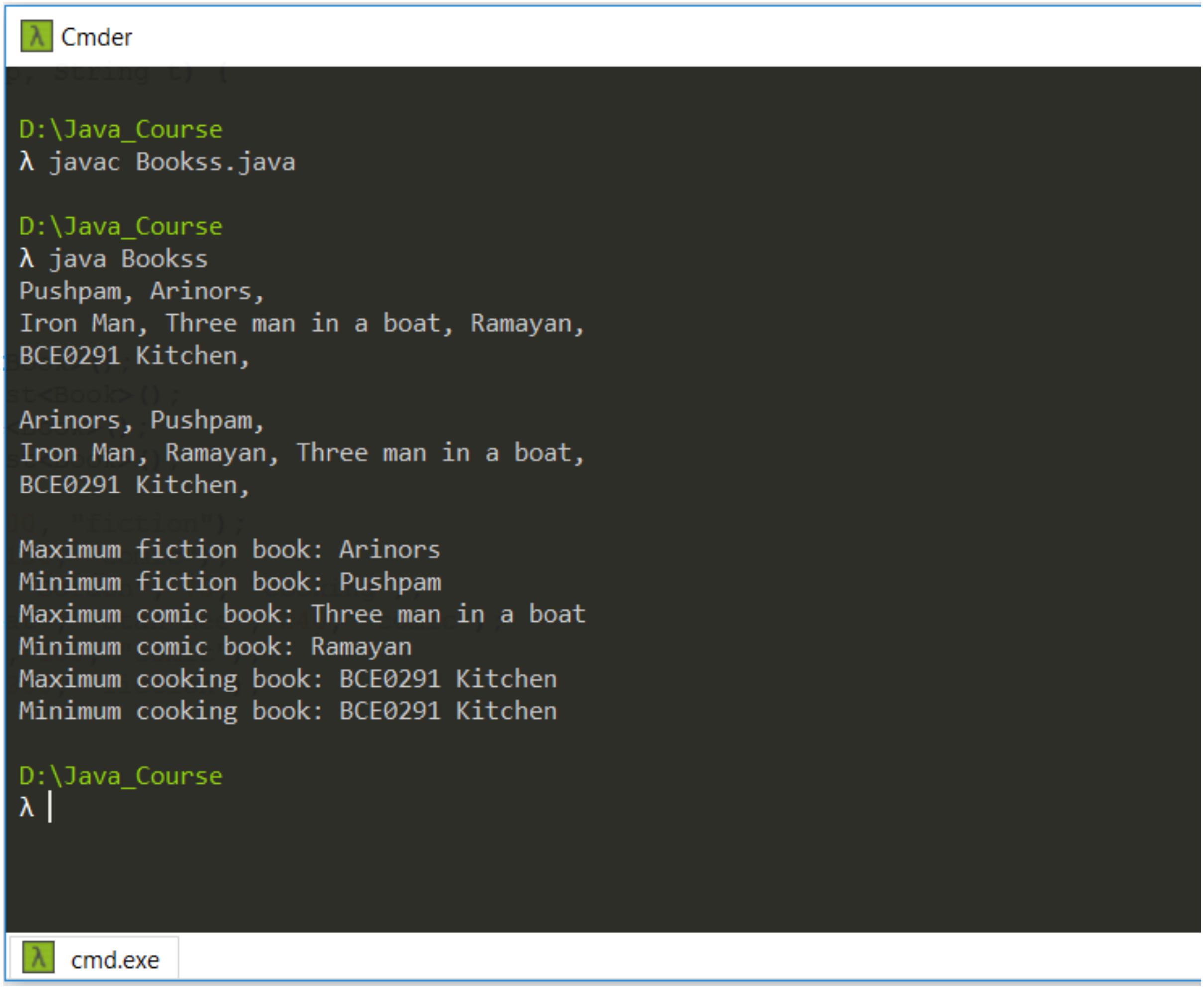
SOURCE CODE:

```
import java.util.*;
import java.lang.*;

public class Bookss {
        public static class Book {
                String book_name;
                String author;
                int price;
                String type;

                public Book(String bn, String a, int p, String t) {
                        this.book_name = bn;
                        this.author = a;
                        this.price = p;
                        this.type = t;
                }
        }

        public static void main(String args[]) {
                ArrayList<Book> list = new ArrayList<Book>();
                ArrayList<Book> fiction = new ArrayList<Book>();
                ArrayList<Book> comic = new ArrayList<Book>();
                ArrayList<Book> cooking = new ArrayList<Book>();
```

```
Book b1 = new Book("Strangers", "JK", 100, "fiction");
Book b2 = new Book("Iron Man", "DC", 150, "comic");
Book b3 = new Book("Hello Kitchen", "Gordon", 70, "cooking");
Book b4 = new Book("Three man in a boat", "Stan Lee", 240, "comic");
Book b5 = new Book("Ramayan", "Amare", 100, "comic");
Book b6 = new Book("Trainors", "JRR", 300, "fiction");

list.add(b1);
list.add(b2);
list.add(b3);
list.add(b4);
list.add(b5);
list.add(b6);

list.forEach((book) -> {
        if(book.type == "fiction") {
                fiction.add(book);
        }
        if(book.type == "comic") {
                comic.add(book);
        }
        if(book.type == "cooking") {
                cooking.add(book);
        }
});

fiction.forEach((n)-> System.out.print(n.book_name+", "));
System.out.println();
comic.forEach((n)-> System.out.print(n.book_name+", "));
System.out.println();
cooking.forEach((n)-> System.out.print(n.book_name+", "));
System.out.println();
System.out.println();

fiction.sort((p1,p2) -> p1.book_name.compareTo(p2.book_name));
comic.sort((p1,p2) -> p1.book_name.compareTo(p2.book_name));
cooking.sort((p1,p2) -> p1.book_name.compareTo(p2.book_name));
fiction.forEach((n)-> System.out.print(n.book_name+", "));
System.out.println();
comic.forEach((n)-> System.out.print(n.book_name+", "));
System.out.println();
cooking.forEach((n)-> System.out.print(n.book_name+", "));
System.out.println();
System.out.println();
```

```
        Book maxFicBook = Collections.max(fiction,Comparator.comparing(b-
>b.price));
        Book minFicBook = Collections.min(fiction,Comparator.comparing(b-
>b.price));
        Book maxComBook =
Collections.max(comic,Comparator.comparing(b->b.price));
        Book minComBook = Collections.min(comic,Comparator.comparing(b-
>b.price));
        Book maxCooBook =
Collections.max(cooking,Comparator.comparing(b->b.price));
        Book minCooBook =
Collections.max(cooking,Comparator.comparing(b->b.price));

        System.out.println("Maximum fiction book: "+
maxFicBook.book_name);
        System.out.println("Minimum fiction book: "+
minFicBook.book_name);
        System.out.println("Maximum comic book: "+
maxComBook.book_name);
        System.out.println("Minimum comic book: "+
minComBook.book_name);
        System.out.println("Maximum cooking book: "+
maxCooBook.book_name);
        System.out.println("Minimum cooking book: "+
minCooBook.book_name);
    }
}
```

EXECUTION:



```
D:\Java_Course
λ javac Bookss.java

D:\Java_Course
λ java Bookss
Pushpam, Arinors,
Iron Man, Three man in a boat, Ramayan,
BCE0291 Kitchen,

Arinors, Pushpam,
Iron Man, Ramayan, Three man in a boat,
BCE0291 Kitchen,

Maximum fiction book: Arinors
Minimum fiction book: Pushpam
Maximum comic book: Three man in a boat
Minimum comic book: Ramayan
Maximum cooking book: BCE0291 Kitchen
Minimum cooking book: BCE0291 Kitchen

D:\Java_Course
λ |
```

RESULT

The list of fiction, comic and cooking of the book is been printed, and shorted order is been printed and the max and the min value of each has been printed

QUESTION 3 :

Write a javafx program to get the students information for selecting the B.E

course Get student name ,age, address, in text field

Get the student 12 th mark in text field (maths phy,che)

Get the course interested in radio button (EEE,CSE,IT,ECE)

Get the sports interested in checkbox (football ,cricket , basketball , Volleyball, Basketball, Tennis)

When the submit button is clicked show a status that "your details is updated" (using event handler)

If the marks is above 100 in each subject "ask the student to give the correct marks" .(using event handler)

BRIEF ABOUT YOUR APPROACH:

We use various functions of JavaFX for getting the input, radio buttons, checkbox and its various parameters

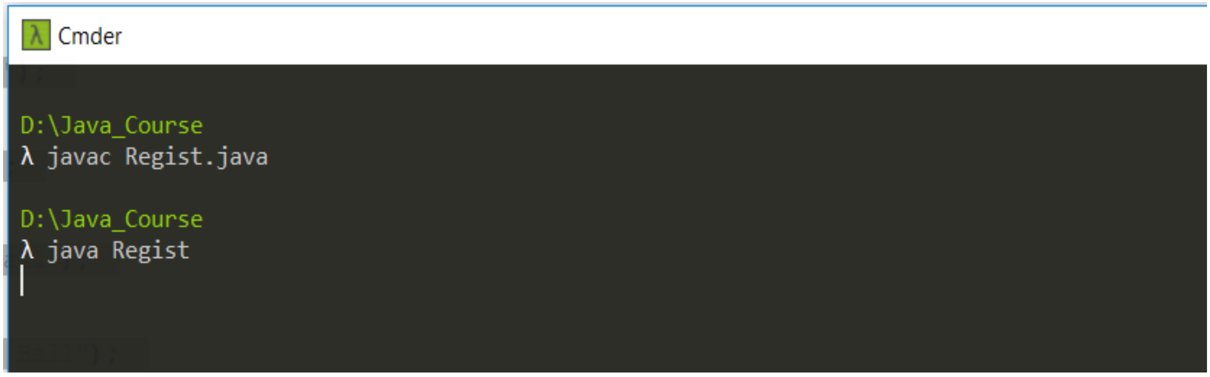
SOURCE CODE:

```
import javafx.application.Application;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.CheckBox;
import javafx.scene.control.ChoiceBox;
import javafx.scene.control.DatePicker;
import javafx.scene.control.ListView;
import javafx.scene.control.RadioButton;
import javafx.scene.layout.GridPane;
import javafx.scene.text.Text;
import javafx.scene.control.TextField;
import javafx.scene.control.ToggleGroup;
import javafx.scene.control.ToggleButton;
import javafx.stage.Stage;

public class Regist extends Application {
  @Override
```

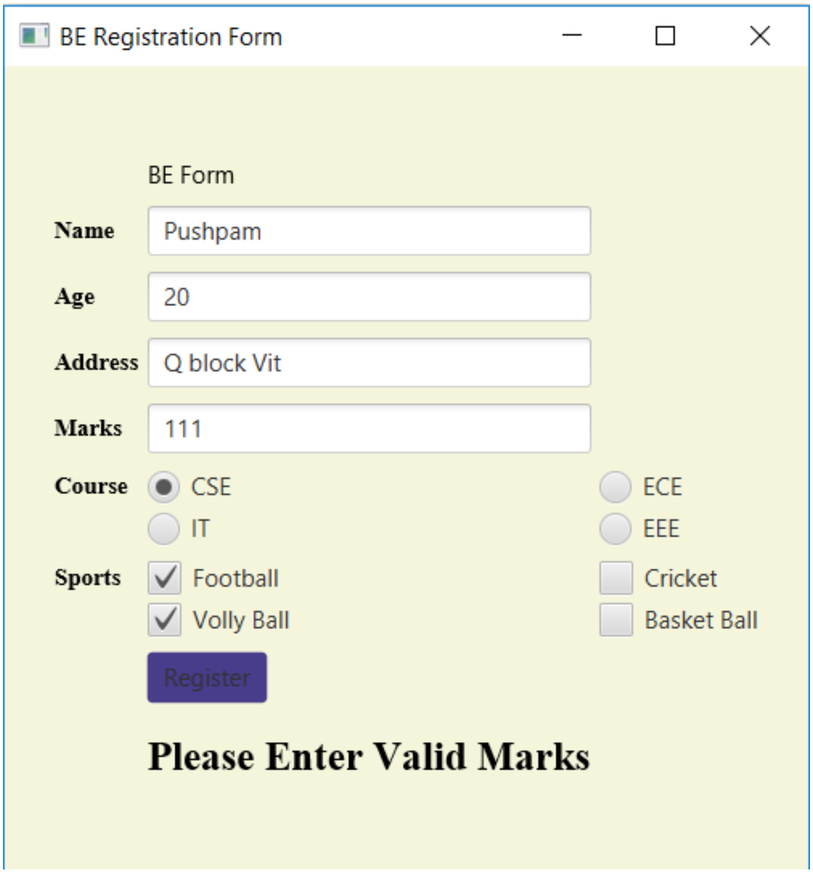
```
public void start(Stage stage) {
  Text heading = new Text("BE Form");
       Text nameLabel = new Text("Name");
  TextField nameText = new TextField();
       Text ageLabel = new Text("Age");
  TextField ageText = new TextField();
       Text addressLabel = new Text("Address");
  TextField addressText = new TextField();
       Text marksLabel = new Text("Marks");
  TextField marksText = new TextField();
  Text courseLabel = new Text("Course");
  ToggleGroup groupCourse = new ToggleGroup();
  RadioButton cseRadio = new RadioButton("CSE");
  cseRadio.setToggleGroup(groupCourse);
  RadioButton eceRadio = new RadioButton("ECE");
  eceRadio.setToggleGroup(groupCourse);
       RadioButton itRadio = new RadioButton("IT");
  itRadio.setToggleGroup(groupCourse);
  RadioButton eeeRadio = new RadioButton("EEE");
  eeeRadio.setToggleGroup(groupCourse);

  Text sportsLabel = new Text("Sports");

  CheckBox footballCheckBox = new CheckBox("Football");
  footballCheckBox.setIndeterminate(false);

  CheckBox cricketCheckBox = new CheckBox("Cricket");
  footballCheckBox.setIndeterminate(false);

       CheckBox vollyballCheckBox = new CheckBox("Volly Ball");
  footballCheckBox.setIndeterminate(false);

       CheckBox basketballCheckBox = new CheckBox("Basket Ball");
  footballCheckBox.setIndeterminate(false);

       Text status = new Text("");

  Button buttonRegister = new Button("Register");

            buttonRegister.setOnAction(new EventHandler<ActionEvent>() {
                public void handle(ActionEvent event) {
                    if(marksText.getText().compareTo("100")<=0 ) {
                        status.setText("Your Details is Updated");

                    }
                    else{
```

```
                                        status.setText("Please Enter Valid Marks");
                                }
                        }
                });

        GridPane gridPane = new GridPane();

        gridPane.setMinSize(500, 500);

        gridPane.setPadding(new Insets(10, 10, 10, 10));

        gridPane.setVgap(5);
        gridPane.setHgap(5);

        gridPane.setAlignment(Pos.CENTER);

        gridPane.add(heading, 1, 0);
            gridPane.add(nameLabel, 0, 2);
        gridPane.add(nameText, 1, 2);

            gridPane.add(ageLabel, 0, 4);
        gridPane.add(ageText, 1, 4);

            gridPane.add(addressLabel, 0, 6);
        gridPane.add(addressText, 1, 6);

                gridPane.add(marksLabel, 0, 8);
        gridPane.add(marksText, 1, 8);
        gridPane.add(courseLabel, 0, 10);
        gridPane.add(cseRadio, 1, 10);
        gridPane.add(eceRadio, 2, 10);
            gridPane.add(itRadio, 1, 11);
        gridPane.add(eeeRadio, 2, 11);
        gridPane.add(sportsLabel, 0, 13);
        gridPane.add(footballCheckBox, 1, 13);
        gridPane.add(cricketCheckBox, 2, 13);
            gridPane.add(vollyballCheckBox, 1, 14);
        gridPane.add(basketballCheckBox, 2, 14);
        gridPane.add(buttonRegister, 1, 16);
            gridPane.add(status, 1, 20);

        buttonRegister.setStyle("-fx-background-color: darkslateblue; -fx-textfill:
    white;");

        nameLabel.setStyle("-fx-font: normal bold 15px 'serif' ");
            ageLabel.setStyle("-fx-font: normal bold 15px 'serif' ");
        addressLabel.setStyle("-fx-font: normal bold 15px 'serif' ");
        marksLabel.setStyle("-fx-font: normal bold 15px 'serif' ");
    courseLabel.setStyle("-fx-font: normal bold 15px 'serif' ");
    sportsLabel.setStyle("-fx-font: normal bold 15px 'serif' ");
        status.setStyle("-fx-font: normal bold 25px 'serif' ");

    gridPane.setStyle("-fx-background-color: BEIGE;");

    Scene scene = new Scene(gridPane);

    stage.setTitle("BE Registration Form");

    stage.setScene(scene);

    stage.show();
  }
  public static void main(String args[]){
    launch(args);
  }
}
```

EXECUTION:



```
D:\Java_Course
λ javac Regist.java

D:\Java_Course
λ java Regist
|
```



BE Form

Name: Pushpam

Age: 20

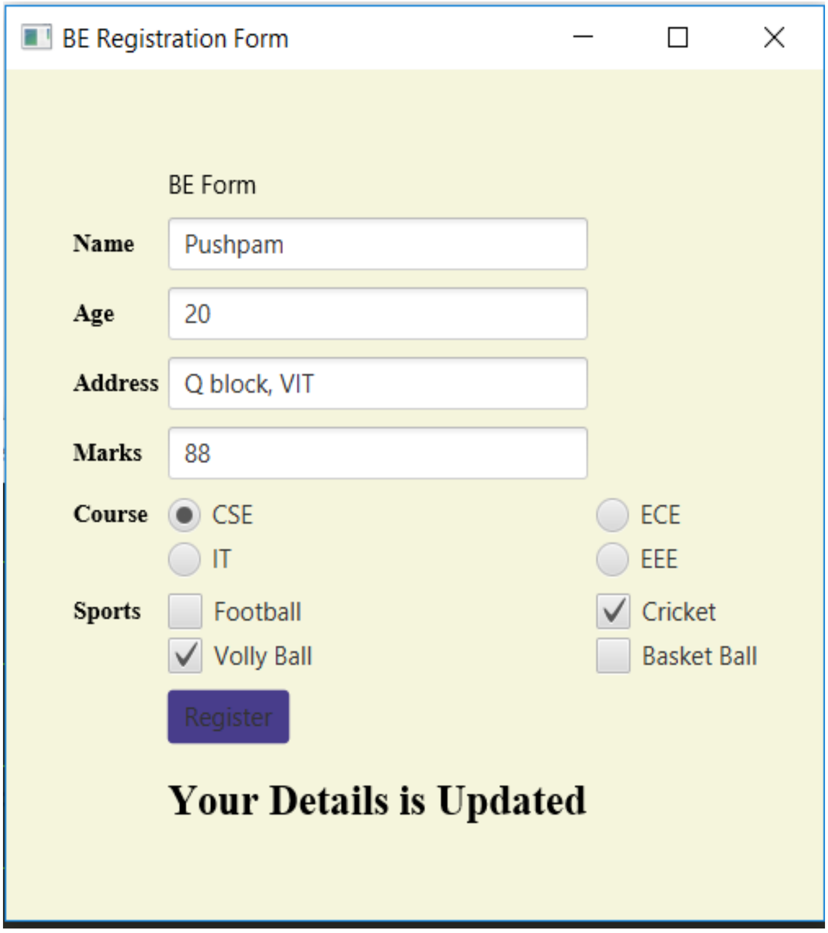
Address: Q block Vit

Marks: 111

Course: ● CSE ○ ECE ○ IT ○ EEE

Sports: ☑ Football ☐ Cricket ☑ Volly Ball ☐ Basket Ball

Please Enter Valid Marks



BE Form

Name: Pushpam

Age: 20

Address: Q block, VIT

Marks: 88

Course: ● CSE ○ ECE ○ IT ○ EEE

Sports: ☐ Football ☑ Cricket ☑ Volly Ball ☐ Basket Ball

Your Details is Updated

RESULT

Form has been displayed, and if the marks are invalid the on click invalid message is shown, else the success message is shown

---

QUESTION 4 :

Write a javafx for mouse event handling

When the mouse is clicked ,screen colours have to change from blue - green –

red Get the x and y coordinates of the mouse when it click in screen area .

Display the status of mouse when mouse enter ,exit of the

screen

BRIEF ABOUT YOUR APPROACH:

We use various functions of JavaFX for mousehandling, mouse click functions to change screen colour and function to get the X & Y coordinates of the mouse when clicked in a certain screen area. We also use a JavaFX Function to display the status of the mouse when we enter/exit from the screen.

SOURCE CODE:

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/*<applet code="NewApplet.class" height="300" width="400">
 </applet>*/
public class NewApplet extends Applet implements
MouseListener,MouseMotionListener
{
 int x,y;
 String str="";
 @Override
 public void start(Stage stage){
          public void init()
 {
 addMouseListener(this);
 addMouseMotionListener(this);
 }
 @Override
```

```
public void paint(Graphics g)
{
g.drawString(str+" x:"+x+" y:"+y, 100, 100);
}

@Override
public void mouseEntered(MouseEvent ae)
{
showStatus("mouse entertd at x:"+x+" y:"+y);
}

@Override
public void mouseExited(MouseEvent ae)
{
showStatus("mouse exited at x:"+x+" y:"+y);
}

@Override
public void mousePressed(MouseEvent ae)
{
setBackground(Color.YELLOW);
}

@Override
public void mouseReleased(MouseEvent ae)
{
setBackground(Color.RED);
}
@Override
public void mouseClicked(MouseEvent ae)
{
x=ae.getX();
y=ae.getY();
repaint();
}
@Override
public void mouseDragged(MouseEvent ae)
{
str="Mouse Dragged";
repaint();
}
@Override
public void mouseMoved(MouseEvent ae)
{
str="Mouse Moved";
repaint();
}

}

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



}
```

EXECUTION:

RESULT