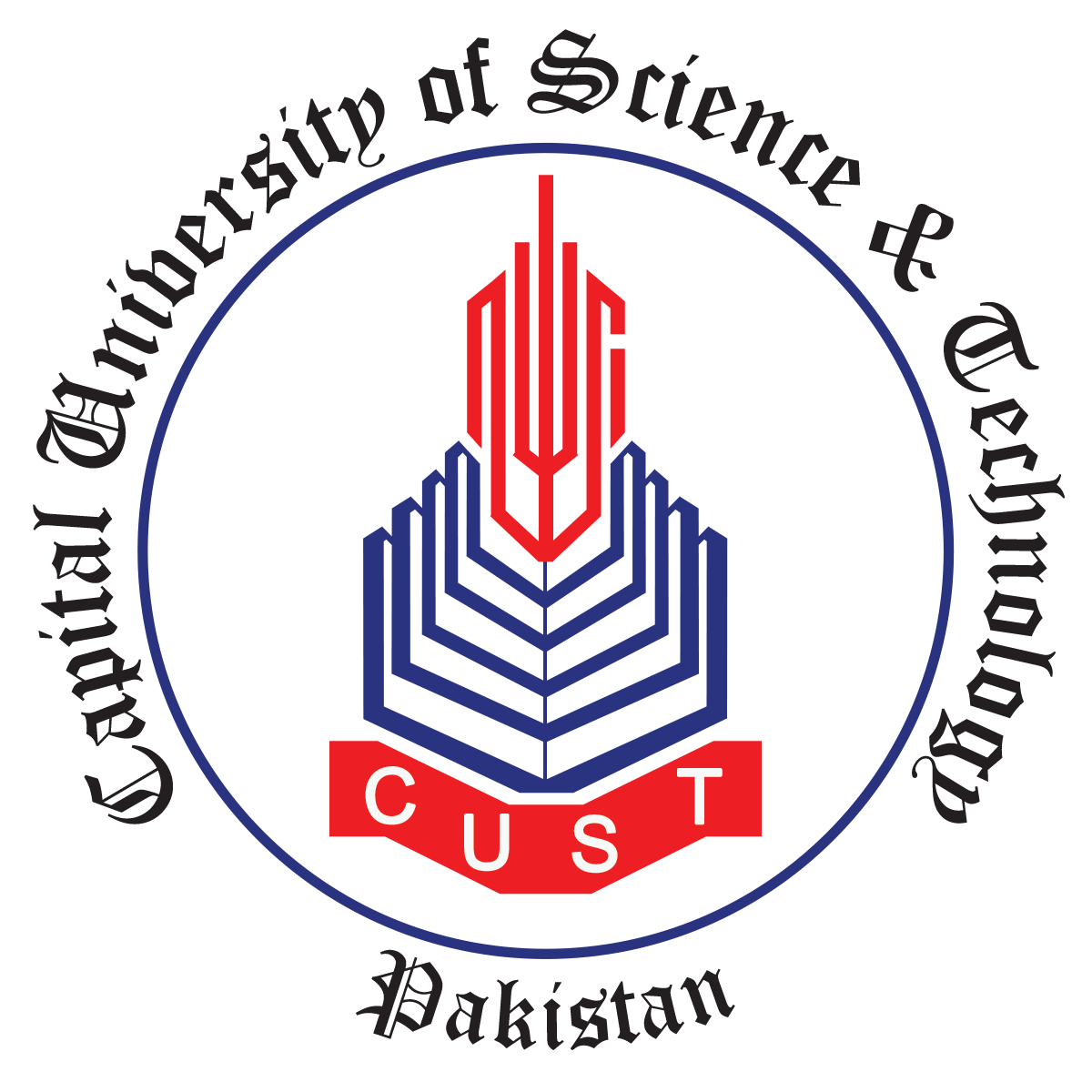
skd laptop

[Email address]



Name: Sunduss Aamir Khan

Registration Number: BSE173047

SOFTWARE TESTING

Section: 2

Assignment 4 and 5

**Full Code:**

public class STassi4\_5 {

boolean checkingtype(int a, int b, double a1, double a2)

{

double ca=0;

ca=180-a1;

if(a1==0 || a2==0)

{

System.out.println("NOT A SHAPE");

return false;

}

else if(a1==180 || a2==1800)

{

System.out.println("NOT A SHAPE");

return false;

}

else if(a==0 || b==0)

{

System.out.println("NOT A SHAPE");

return false;

}

else if(a==b && a!=0 && b!=0)

{

if(a1==a2 && a1==90)

{

System.out.println("SQUARE");

return true;

}

else if(a1<a2 || a1>a2)

{

if(a1!=90 && a2!=90)

{

if(ca==a2)

{

System.out.println("RHOMUS");

return true;

}

else

{

System.out.println("NOT A SHAPE");

return false;

}

}

else if( a1==90 || a2==90)

{

System.out.println("NOT A SHAPE");

return false;

}

else

{

System.out.println("NOT A SHAPE");

return false;

}

}

else

{

System.out.println("NOT A SHAPE");

return false;

}

}

else

{

System.out.println("NOT A SHAPE");

return false;

}

}

public static boolean main(String[] args)

{

STassi4\_5 quadrilaterial = new STassi4\_5();

double a1=90;

double a2=90;

double c=0;

//swapping greater angle to a1

if(a1<a2)

{

c=a1;

a1=a2;

a2=c;

}

int h=10;

int w=10;

if (quadrilaterial.checkingtype(h,w, a1, a2))

return (true);

else

return (false);

}

}

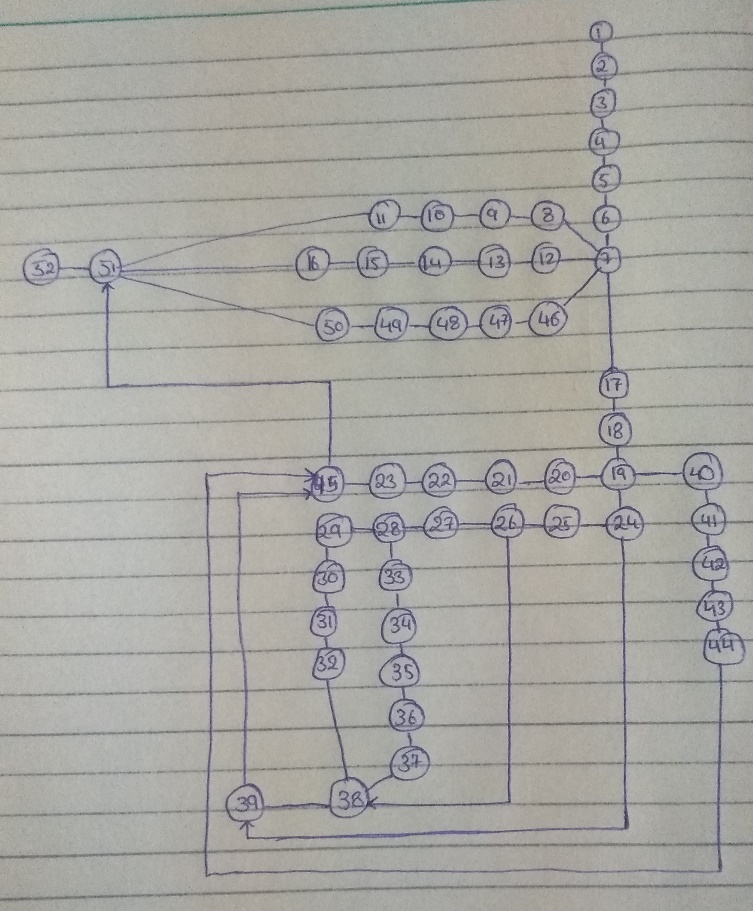
**Selected portion of the code:**

|  |  |
| --- | --- |
|  |  |
| 1. | public class TestingAssi2 |
| 2. | { |
| 3. | boolean checkingtype(int a, int b, double a1, double a2) |
| 4. | { |
| 5. | double ca=0; |
| 6. | ca=180-a1; |
| 7. | if(a1==0 || a2==0) |
| 8. | { |
| 9. | System.out.println("NOT A SHAPE"); |
| 10. | return false; |
| 11. | } |
| 12. | else if(a1==180 || a2==180 || a==0 || b==0) |
| 13. | { |
| 14. | System.out.println("NOT A SHAPE"); |
| 15. | return false; |
| 16. | } |
| 17. | else if(a==b && a!=0 && b!=0) |
| 18. | { |
| 19. | if(a1==a2 && a1==90) |
| 20. | { |
| 21. | System.out.println("SQUARE"); |
| 22. | return true; |
| 23. | } |
| 24. | else if(a1<a2 || a1>a2) |
| 25. | { |
| 26. | if(a1!=90 && a2!=90) |
| 27. | { |
| 28. | if(ca==a2) |
| 29 | { |
| 30. | System.out.println("RHOMUS"); |
| 31. | return true; |
| 32. | } |
| 33. | else |
| 34. | { |
| 35. | System.out.println(“NOT A SHAPE"); |
| 36. | return false; |
| 37. | } |
| 38. | } |
| 39. | } |
| 40. | else |
| 41. | { |
| 42. | System.out.println(“NOT A SHAPE"); |
| 43. | return false; |
| 44. | } |
| 45. | } |
| 46. | else |
| 47. | { |
| 48. | System.out.println(“NOT A SHAPE"); |
| 49. | return false; |
| 50. | } |
| 51. | } |
| 52. | } |

**MCDC**

1. a1=0, a2=0, a=0, b=0, ca=180 🡪 Not a shape (False)
2. a1=10, a2=0, a=0, b=0, ca=170 🡪 Not a shape (False)
3. a1=0, a2=10, a=0, b=0, ca=180 🡪 Not a shape (False)
4. a1=0, a2=0, a=10, b=0, ca=180 🡪 Not a shape (False)
5. a1=10, a2=0, a=0, b=10, ca=180 🡪 Not a shape (False)
6. a1=180, a2=180, a=0, b=0, ca=0 🡪 Not a shape (False)
7. a1=180, a2=180, a=10, b=0, ca=0 🡪 Not a shape (False)
8. a1=180, a2=180, a=0, b=10, ca=0 🡪 Not a shape (False)
9. a1=180, a2=90, a=0, b=0, ca=0 🡪 Not a shape (False)
10. a1=90, a2=180, a=0, b=0, ca=90 🡪 Not a shape (False)
11. a1=90, a2=80, a=10, b=10, ca=90 🡪 Not a shape (False)
12. a1=90, a2=90, a=10, b=10, ca=90 🡪 Square (True)
13. a1=80, a2=90, a=20, b=20, ca=100 🡪 Not a shape (False)
14. a1=90, a2=80, a=20, b=20, ca=90 🡪 Not a shape (False)
15. a1=60, a2=120, a=20, b=20, ca=120 🡪 Rhombus (True)
16. a1=120, a2=60, a=20, b=20, ca=60 🡪 Rhombus (True)
17. a1=90, a2=90, a=0, b=0, ca=0 🡪 Not a shape (False)
18. a1=80, a2=90, a=0, b=0, ca=100 🡪 Not a shape (False)
19. a1=90, a2=80, a=0, b=0, ca=90 🡪 Not a shape (False)
20. a1=60, a2=120, a=0, b=0, ca=120 🡪 Not a shape (False)
21. a1=120, a2=60, a=0, b=0, ca=60 🡪 Not a shape (False)

**Path Prediction Expressions**



1. 1🡪2🡪3🡪4🡪5🡪6🡪7🡪8🡪9🡪10🡪11🡪51🡪52

a1=0, a2=0, a=0, b=0, ca=180

1. 1🡪2🡪3🡪4🡪5🡪6🡪7🡪12🡪13🡪14🡪15🡪16🡪51🡪52

a1=180, a2=180, a=0, b=0, ca=0

1. 1🡪2🡪3🡪4🡪5🡪6🡪7🡪46🡪47🡪48🡪49🡪50🡪51🡪52

a1=90, a2=80, a=10, b=10, ca=90

1. 1🡪2🡪3🡪4🡪5🡪6🡪7🡪17🡪18🡪19🡪20🡪21🡪22🡪23🡪45🡪51🡪52

a1=90, a2=90, a=10, b=10, ca=90

1. 1🡪2🡪3🡪4🡪5🡪6🡪7🡪17🡪18🡪19🡪24🡪25🡪26🡪38🡪39🡪45🡪51🡪52

a1=60, a2=20, a=20, b=20, ca=120

1. 1🡪2🡪3🡪4🡪5🡪6🡪7🡪17🡪18🡪19🡪24🡪25🡪26🡪27🡪28🡪33🡪34🡪35🡪36🡪3🡪38🡪39🡪45🡪51🡪52

a1=60, a2=120, a=20, b=20, ca=120

1. 1🡪2🡪3🡪4🡪5🡪6🡪7🡪17🡪18🡪19🡪24🡪25🡪26🡪27🡪28🡪29🡪30🡪31🡪32🡪38🡪39

🡪45🡪51🡪52

a1=120, a2=60, a=20, b=20, ca=60

1. 1🡪2🡪3🡪4🡪5🡪6🡪7🡪17🡪18🡪19🡪24🡪45🡪51🡪52

a1=20, a2=20, a=20, b=20, ca=160

1. 1🡪2🡪3🡪4🡪5🡪6🡪7🡪17🡪18🡪19🡪40🡪41🡪42🡪43🡪44🡪45🡪50🡪51

a1=20, a2=20, a=20, b=20, ca=160

**Test Oracle**

To verify the execution is correct, we need to compare the actual outcome with the expected outcome. Test oracle is a tool that can return the expected outcome for a given input test paths.

1. 1🡪2🡪3🡪4🡪5🡪6🡪7🡪8🡪9🡪10🡪11🡪51🡪52

a1=0, a2=0, a=0, b=0, ca=180

Expected outcome: Not a shape

Actual outcome: Not a shape

1. 1🡪2🡪3🡪4🡪5🡪6🡪7🡪12🡪13🡪14🡪15🡪16🡪51🡪52

a1=180, a2=180, a=0, b=0, ca=0

Expected outcome: Not a shape

Actual outcome: Not a shape

1. 1🡪2🡪3🡪4🡪5🡪6🡪7🡪46🡪47🡪48🡪49🡪50🡪51🡪52

a1=90, a2=80, a=10, b=10, ca=90

Expected outcome: Not a shape

Actual outcome: Not a shape

1. 1🡪2🡪3🡪4🡪5🡪6🡪7🡪17🡪18🡪19🡪20🡪21🡪22🡪23🡪45🡪51🡪52

a1=90, a2=90, a=10, b=10, ca=90

Expected outcome: Square

Actual outcome: Square

1. 1🡪2🡪3🡪4🡪5🡪6🡪7🡪17🡪18🡪19🡪24🡪25🡪26🡪38🡪39🡪45🡪51🡪52

a1=60, a2=20, a=20, b=20, ca=120

Expected outcome: Not a shape

Actual outcome: Not a shape

1. 1🡪2🡪3🡪4🡪5🡪6🡪7🡪17🡪18🡪19🡪24🡪25🡪26🡪27🡪28🡪33🡪34🡪35🡪36🡪3🡪38🡪39🡪45🡪51🡪52

a1=60, a2=120, a=20, b=20, ca=120

Expected outcome: Rhombus

Actual outcome: Rhombus

1. 1🡪2🡪3🡪4🡪5🡪6🡪7🡪17🡪18🡪19🡪24🡪25🡪26🡪27🡪28🡪29🡪30🡪31🡪32🡪38🡪39

🡪45🡪51🡪52

a1=120, a2=60, a=20, b=20, ca=60

Expected outcome: Rhombus

Actual outcome: Rhombus

1. 1🡪2🡪3🡪4🡪5🡪6🡪7🡪17🡪18🡪19🡪24🡪45🡪51🡪52

a1=20, a2=20, a=20, b=20, ca=160

Expected outcome: Not a shape

Actual outcome: Not a shape

1. 1🡪2🡪3🡪4🡪5🡪6🡪7🡪17🡪18🡪19🡪40🡪41🡪42🡪43🡪44🡪45🡪50🡪51

a1=20, a2=20, a=20, b=20, ca=160

Expected outcome: Not a shape

Actual outcome: Not a shape