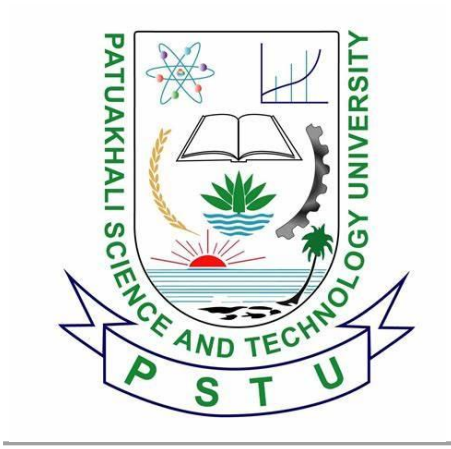

PATUAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY

COURSE CODE CCE-122

SUBMITTED TO:

Sarna Majumder

Department of Computer and Communication Engineering
Faculty of Computer Science and Engineering

SUBMITTED BY:

Md. Sharafat Karim

ID: **2102024**,

Registration No: **10151**

Faculty of Computer Science and Engineering

Date of submission: **7 January, 2024**

Assignment no: 04

Assignment title: instanceof operator

Md. Sharafat Karim
ID: 2102024
Reg: 10151

Topic: Instanceof operator

Questions:

Total 5

(1) How is instanceof operator different from "==" in java? And in what scenarios would you prefer using one over another? (2)

(2) While working with multiple class hierarchy involving inheritance, what kind of issues may arrise if we use instanceof operator? Explain with a proper example. (3)

Answers: (Topic: Instanceof operator)

(1) How is instanceof operator different from "==" in java? And in what scenarios would you prefer using one over another? (2)

Ans: In java "==" operator is used for the comparison of reference. It checks if two reference point to the exact same location. On the other hand instanceof operator is used for type checking, which determines if an object is an instance of a particular class or interface.

"==" operators can be used when we are compairing primitive types, or when we want to explicitly compare object references. On the other hand, instanceof can be used to perform downcasting and polymorphism based operations where we need specific data type.

(2) While working with multiple class hierarchy involving inheritance, what kind of issues may arrise if we use instanceof operator? Explain with a proper example. (3)

Ans: The instanceof operator considers subclasses as instances of their parent class, which may lead to unintended behavior when checking for specific types.

To illustrate this problem, lets consider an inheritance hierarchy where a class Shape is extended to Circle, which is extended to Square. Now if we create an object of child class then instanceof operator will return 1 for every class of the hierarchy which might cause a problem if we want to determine a certain class specifically.

```
class Shape {}

class Circle extends Shape {}

class Square extends Circle {}

public class Test {
    public static void main (String [] args) {
        Shape shape = new Square();

        if (shape instanceof Square) {
            System.out.println("Square");
        }
        if (shape instanceof Circle) {
            System.out.println("Circle");
        }
        if (shape instanceof Shape) {
            System.out.println("Shape");
        }
    }
}
```

Here instanceof operator will print ~~both~~ three. So in this way we can't specifically determine a certain class.