



Submitted By:

Hamza Riaz (12282)

BS (SE) Section: A

Submitted To:

Sir ayub

Dated: 29 June 2020

Department of Computer Science,

Department of Computer Science

Final Term Exam Spring 2020

National University, Peshawar Pakistan

Subject: Object Oriented Programming

BS (CS,SE)

Instructor: M.Ayub Khan

There are total 5 questions in this paper.

Max Marks: 50

Note:

At the top of the answer sheet there must be the ID, Name and semester of the concerned Student.

Students must have to provide the output of their respective programs. Students have same answers or programs will be considered fail. Programs in Java or codes should be explained clearly.

As this paper is online so incase of any ambiguity my Whatsapp no. is 034499121116.

**Each question carry equal marks.
Please answer briefly.**

- Q1. a. Why access modifiers are used in java, explain in detail Private and Default access modifiers?
b. Write a specific program of the above mentioned access modifiers in java.
- Q2. a. Explain in detail Public and Protected access modifiers?
b. Write a specific program of the above mentioned access modifiers in java.
- Q3. a. What is inheritance and why it is used, discuss in detail ?
b. Write a program using Inheritance class on Animal in java.
- Q4. a. What is polymorphism and why it is used, discuss in detail ?
b. Write a program using polymorphism in a class on Employee in java.
- Q5. a. Why abstraction is used in OOP, discuss in detail ?
b. Write a program on abstraction in java.

Question No 1 :

Answer:

Access modifiers are the integral part of object-oriented programming. Access modifiers are used to execute encapsulation of OOP. Access modifiers allow you to define who does or who doesn't access to certain features.

Private access modifier:

The scope of private access modifier is limited up to class only:

- 1) Private data members can be accessible within the class.**
- 2) Class and interface cannot be declared as private.**
- 3) If a class has private constructor, then you cannot create the object of that class from outside the class.**

Default access modifier: When something is not mentioned in access modifier is known as Default access modifier. This modifier is limited to the package only. This means that if we have a class with the default access modifier in a package, only those classes in this package can access this class.

```
b) class ABC{  
    private double num = 100;  
    private int square(int a){  
        return a*a;  
    }  
}  
public class Example{  
    public static void main(String args[]){  
        ABC obj = new ABC();
```

```
System.out.println(obj.num);
System.out.println(obj.square(10));
}
}
```

Output:

Compile - time error

Answer No 2:

Public Access modifier: The class method can be declared as public it means that it is accessible from anywhere in the program. Public access modifier is just like public school where everyone can seek admission.

For example:

```
public class Animal {
}
```

Protected Access modifier: The protected access modifier is that which is specified by the key protected. Protected access modifier is just as default access modifier. You can also say that the protected

access modifier is similar to default access modifier with one exception that it has visibility in sub classes. It is only used in parent child relationship.

```
b)  
public class Addition {  
  
    protected int addTwoNumbers(int a, int b){  
        return a+b;  
    }  
}
```

```
import abcpackage.*;  
package xyzpackage;  
class Test extends Addition{  
    public static void main(String args[]){  
        Test obj = new Test();  
        System.out.println(obj.addTwoNumbers(11, 22));  
    }  
}
```

Output:

33

Answer no 3:

Inheritance: Inheritance is that when one object acquires all the properties and behavior of a parent object is known as inheritance. There are three types of inheritance.

- 1) Single level inheritance**
- 2) Multilevel inheritance**
- 3) Hierarchal inheritance**

Inheritance is use is the reusability of code. It means that the code present in the parent class doesn't need to be written again in child class.

The second use is method overriding.

b)

```
public class Animal {
```

```
    private boolean vegetarian;
```

```
    private String eats;
```

```
private int noOfLegs;

public Animal(){}

public Animal(boolean veg, String food, int legs){
    this.vegetarian = veg;
    this.eats = food;
    this.noOfLegs = legs;
}

public boolean isVegetarian() {
    return vegetarian;
}

public void setVegetarian(boolean vegetarian) {
    this.vegetarian = vegetarian;
}

public String getEats() {
    return eats;
}

public void setEats(String eats) {
    this.eats = eats;
}
```

```
    }  
  
    public int getNoOfLegs() {  
        return noOfLegs;  
    }  
  
    public void setNoOfLegs(int noOfLegs) {  
        this.noOfLegs = noOfLegs;  
    }  
  
}
```

Answer no 4:

Polymorphism: If one task is performed in different ways. It is known as polymorphism The most common use of polymorphism in OOP occurs when a parent class reference is used to refer to a child class object.

The reason is that polymorphism is needed in java because the concept is extensively used in implementing inheritance. It plays very

important role in allowing objects having different internal structures to share same external interface.

```
b) public class Employee {  
    private String name;  
    private String address;  
    private int number;
```

```
    public Employee(String name, String address, int number) {  
        System.out.println("Constructing an Employee");  
        this.name = name;  
        this.address = address;  
        this.number = number;  
    }
```

```
    public void mailCheck() {  
        System.out.println("Mailing a check to " + this.name + " " +  
this.address);  
    }
```

```
    public String toString() {  
        return name + " " + address + " " + number;  
    }
```

```
    public String getName() {  
        return name;  
    }
```

```
    public String getAddress() {  
        return address;  
    }
```

```
    public void setAddress(String newAddress) {  
        address = newAddress;  
    }
```

```
public int getNumber() {  
    return number;  
}
```

Answer no 5:

Abstraction: The data which is selected from a larger pool data to show the relevant details of the object to the user. It shows the essential attributes and hides unnecessary information. It also helps to reduce programming complexity and effort.

It is the key element of good software design. It helps in encapsulate behavior and decouple software elements. It helps to have more self-contained modules.

```
abstract class Animal {  
  
    public abstract void animalSound();  
  
    public void sleep() {  
        System.out.println("Zzz");  
    }  
}  
  
class Pig extends Animal {  
    public void animalSound() {  
  
        System.out.println("The pig says: wee wee");  
    }  
}
```

```
}
```

```
class MyMainClass {
```

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

```
    Pig myPig = new Pig(); // Create a Pig object
```

```
    myPig.animalSound();
```

```
    myPig.sleep();
```

```
}
```

```
}
```