

Name :- Amir Zaman

Class :- BS (Computer Science)
2nd semester

Subject :- fundamental
Programming

Roll No :- 16390

Question #5-

Part # A

Why abstraction is used
in oop, discuss in detail?

Answer:-

Abstraction in oop:-

Abstraction is selecting data from a larger pool to show only the relevant detail of the object to the relevant user. Abstraction "shows" only the essential attributes and "hides" unnecessary information. It helps to reduce programming complexity and effort. It is one of the most important concepts of oops.

Advantages of Abstractions -

* The main benefit of using an abstract class is that it allows you to group several classes as siblings.

* Abstractions help to reduce the complexity of the design and implementational process of software.

Summary -

Abstraction is the process of selecting important data sets for an object in your software and leaving out the insignificant ones.

* Once you have modeled your object using

Abstraction, the same set of data could be used in different applications.

* -

part # B
write a program on
abstraction in Java?

Answer:

Abstraction in Java:-

Abstraction is the property
by virtue of which
only the essential
detail are displayed
to the user. The
trivial or the non-
essentials are
not displayed to the
user.

Programs:-

lets implement the
same and create an
interface called Spoot
Java.

```
public interface spoot {  
    void play ();  
}
```

// Now we will create
class name "Badminton"
public class Badminton
implements spoot {

```
public void play () {  
    System.out.println ("playing  
    Badminton");  
}
```

// Next : let's create our
last class "football"
public class Football
implements sport {

```
public void play () {  
    System.out.println ("Playing  
    football");  
}
```

The last step is to
create a main class
named "up.sportInterface".

Question # 4

Part # A

What is polymorphism
and why it is used
- ✓ circus in + daily

Answers

Polymorphism:-

the ability of an object to take on many forms. The most common use of

polymorphism in oop occurs when a parent class reference is used to refer to a child class object.

Any Java object that can pass more than one is-A test.

is considered to a polymorphic. In Java all Java object are

polymorphic since any object will pass the

is-A test for the type one for the class object.

Use of Polymorphism:
is one of the features that allow us to perform a single action in different ways.

Example:-

Let's say we have a class Animal that has a method sound(). Since this is a generic class so we can't give it a implementation like Roar, meow, etc. we had to give a generic message.

```
public class Animal {  
    ...  
    public void sound () {  
        System.out.println("Animal is making a sound");  
    }  
}
```

Question # 4

Part # B

write a program
using polymorphism in
a class on employee
in Java?

Answer,

1* File name : Employee.java

```
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 mail check () {  
        System.out.println ("mailing a  
        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 ;  
}
```

```
{
```

```
}
```


Question # 3

Part # A

what is inheritance and why it is used : discuss in detail?

Answer:-

Inheritance:-

Inheritance is a mechanism in which one class acquires the properties of another class.

Examples:-

A child inherits the traits of his/her parent. With inheritance we can reuse the fields and methods of the existing class. Hence, inheritance facilitates reusability and is an important concept of oops.

Types:-

- ① single inheritance
- ② multiple inheritance
- ③ multilevel inheritance
- ④ hierarchical inheritance
- ⑤ hybrid inheritance

use of inheritance:-

of the most important
concept in object-oriented
programming is that
us to define a class
in terms of another class
which makes it easier to
create and maintain
an opportunity to reuse
the code functionality
and fast implementation
time.



when creating a class
instead of writing completely
new data members and member
function, the programmer
can designate that the new
class should inherit the member
of an existing class.

Question # 3

Part # B

write a program using inheritance class on Animal in Java:

Answer:

~~Page from Java~~

When a class inherits another class it is known as a single inheritance.

Programs:-

```
class Animal {  
void eat () { System.out.println("eating ..."); }  
}
```

```
class Dog extends Animal {  
void bark () { System.out.println("barking ..."); }  
}
```

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

```
Dog d = new Dog ();
```

```
d. bark ();  
d. eat ();  
};
```

outputs:

barking - - -
eating - - -

Question # 2

Part # A

Explain in detail public
and protected access
modifiers?

Answer:

Protected Access modifiers.

protected data
members and method are
only accessible by the
classes of the
same package and subclasses
present in any package
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.

Example:-

In this example the
class Test which is present
in another package is
able to call the
add Two Number method.

```

package abc package ;
public class void main ()
String [] arguments ) {
{

```

Public Access Modifier :-

Class, method, constructor, interface etc. declared public can be accessed from any other class. Therefore fields, methods block declared inside a public class can be accessed from any class belonging to the java universe.

Example :-

Uses the following function with public access control.

```

public static void main
(String [] arguments) {
// ...
}

```

Question # 2

Part # B

Protected:-

Program:-

```
package mypack;  
import pack.*;
```

```
class B extends A {
```

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

```
        B obj = new B();
```

```
        obj.msg();
```

```
    }
```

```
}  
Output:- Hello
```

Public:-

Program:-

```
// save by A.java
```

```
package pack;
```

```
public class A {
```

```
    public void msg ()
```

```
    {  
        out.println ("Hello");  
    }  
}
```

```
} system
```

```
// Save by B Java
```

```
package mypackage  
import package.*;
```

```
class B {
```

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

```
A obj = new A ();
```

```
obj . msg ();
```

```
}
```

```
}
```

Output :-

Hello.

Question # 1

Part # A

Why are access modifiers used in Java, explain in detail. Private access modifiers? Default access modifiers?

Answer

Access Modifier in Java:

The access modifiers in Java specify the accessibility of a field, method, constructor, or class. We can change the access level of field.

Private Access modifier:-

The private access modifier is accessible only within the class.

Examples:-

We have created two classes A and Simple. Class A contains private data member and private method.

Role of private constructor

If you make any class constructor private you cannot create the instance of that class from outside the class.

Examples -

```
class A {
    private A () {} // private constructor

    void msg () { system.out.println("Hello Java"); }
}

public class Simple {
    public static void main (String args []) {
        A obj = new A () // compile time error
    }
}
```

Default

If you don't use any modifier, it is treated as default by the compiler. The default modifier is accessible.

from outside the package
it provides more accessibility
then private restricted, but it is
more protected and then public

Part # B

Private access program

class A {

private A() {} // private
Constructors

void msg () { System.out.println
("Hello Java"); }

public class simple {
public static void main
(String args[]) {

A obj = new A(); // compile
Time error
}

Default.

// Save by A Java

```
Package pack;  
Class A {  
void msg () {System.out.println  
("Hello");}  
}
```

Save by B Java

```
Package mypack;  
import pack.*;  
Class B {  
public static void main  
(String args[]) {
```

```
A obj = new A (); // compile  
Time 8800
```

```
obj msg (); // compile Time 8800  
}  
}
```