SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NAAC – UGC with ‘A’ Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

23CSB101
OBJECT ORIENTED PROGRAMMING

UNIT II
INHERITANCE, PACKAGES AND INTERFACES

METHOD OVERRIDING

By
M.Kanchana

Assistant Professor/CSE

MCQ

What happens if we don’t override an abstract method in a subclass?

A) The program will run without any issues

B) The subclass will remain abstract and must be declared as abstract

C) The compiler will generate a warning but allow compilation

D) The subclass can be instantiated normally

MCQ

What happens if we don’t override an abstract method in a subclass?

A) The program will run without any issues

B**) The subclass will remain abstract and must be declared as abstract**

C) The compiler will generate a warning but allow compilation

D) The subclass can be instantiated normally

MCQ

What is the key characteristic of an abstract class in Java?
A) It cannot contain any concrete methods
B) It must have at least one abstract method
C) It can be instantiated directly
D) It cannot have constructors

MCQ

What is the key characteristic of an abstract class in Java?
A) It cannot contain any concrete methods
B) It must have at least one abstract method
C) It can be instantiated directly
D) It cannot have constructors

MCQ

What will be the output of the following code?

```
abstract class Animal {
    void eat() {
        System.out.println("Animals eat food");
    }
    abstract void sound();
}

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

```
public class Main {
    public static void main(String[] args) {
        Animal obj = new Dog();
        obj.eat();
        obj.sound();
    }
}
```

MCQ

OUTPUT

```
Animals eat food
Dog barks
```

MCQ

```
class A {
    void show() {
        System.out.println("Class A method");
    }
}

class B extends A {
    void show() {
        System.out.println("Class B method");
    }
}
```

```
class C extends B {
    void show() {
        System.out.println("Class C method");
    }
}

public class Main {
    public static void main(String[] args) {
        A obj = new C();
        obj.show();
    }
}
```

MCQ

OUTPUT

Class C method

MCQ

```
class Exam {
    final void rules() {
        System.out.println("Students must follow
exam rules");
    }

class OnlineExam extends Exam {
    void rules() {
 System.out.println("Online exam has different
rules"); }
}

public class Main {
    public static void main(String[] args) {
        Exam obj = new Exam();
        obj.rules();
    }
}
```

Will it print output or error?

MCQ

```
abstract class Animal {
    Animal() {
        System.out.println("Animal is created");
    }
    abstract void makeSound();
}

class Dog extends Animal {
    Dog() {
        System.out.println("Dog is created");
    }
    void makeSound() {
        System.out.println("Woof Woof");
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        Animal obj = new Dog();
        obj.makeSound();
    }
}
```

MCQ

OUTPUT:

```
Animal is created
Dog is created
Woof Woof
```

MCQ

```
class Parent {
    static void show() {
        System.out.println("Parent class static method");
    }
}
class Child extends Parent {
    static void show() {
        System.out.println("Child class static method");
    }
}
public class Main {
    public static void main(String[] args) {
        Parent obj = new Child();
        obj.show();
    }
}
```

MCQ

A) Prints: Child class static method
B) Prints: Parent class static method
C) Compilation error
D) Runtime error

Static methods are not overridden, but hidden. Method calls depend on reference type (Parent obj), not runtime object (Child).

THANK YOU