MSSV:51900491

Tên :Cao Nguyễn Kỳ Duyên

**8. Abstract Class exercises**

**1/Can you create instance of abstract class?**

**Answer:** No, you cannot create an instance of an abstract class because it does not have a complete implementation. The purpose of an abstract class is to function as a base for subclasses. It acts like a template, or an empty or partially empty structure, you should extend it and build on it before you can use it.

**2/ Is it necessary for abstract class to have abstract method?**

**Answer**: It's not necessary for an abstract class to have abstract method. ... Java Abstract class can implement interfaces without even providing the implementation of interface methods. Java Abstract class is used to provide common method implementation to all the subclasses or to provide default implementation.

**3/** **Which of the following is FALSE about abstract classes in Java?**

**Answer**: (C) A class can be made abstract without any abstract method

**4/ Predict the output of the following program.**

**Answer**: Compilation error

**5/ How to achieve abstraction in java?**

**Answer** Abstraction in Java or Object oriented programming is a way to segregate implementation from an interface and one of the five fundamentals along with Encapsulation, Inheritance, Polymorphism, Class, and Object. Abstraction in Java is achieved by using interface and abstract class in Java.

**6/** **Which of the following classes would be suitable as an abstract class?**

**Answer** d) Shape

**7/ Predict the output of the following program.**

**Answer:** The pig says: wee wee

Zzz

**8/ Predict the output of the following program.**

**Answer** Base Constructor Called

Derived Constructor Called

**9. Write an abstract class Shape**

**– Data members: numSides**

**– Constructor: initialize numSides**

**– Concrete method: get method for numSides**

**– Abstract methods: getArea(), getPerimeter()**

**Answer**

public abstract class Shape

{

private int numSides;

public Shape( int newSides)

{

numSides = newSides;

}

public int getNumSides()

{

return numSides;

}

public abstract double getArea();

public abstract double getPerimeter();

}