

Assignment – classes and inheritance

Priyanshu Jhansla

88008

Ques 1.

(a).

```
class A
{
    public A()
    {
        System.out.println("Class A Constructor");
    }
}
class B extends A
{
    public B()
    {
        System.out.println("Class B Constructor");
    }
}
class C extends B
{
    public C()
    {
        System.out.println("Class C Constructor");
    }
}
public class MainClass
{
    public static void main(String[] args)
    {
        C c = new C();
    }
}
```

Ans. Class A Constructor

Class B Constructor

Class C Constructor

(b).

```
class A
{
    String s = "Class A";
}
```

```

class B extends A
{
    String s = "Class B";
    {
        System.out.println(super.s);
    }
}
class C extends B
{
    String s = "Class C";
    {
        System.out.println(super.s);
    }
}
public class MainClass
{
    public static void main(String[] args)
    {
        C c = new C();
        System.out.println(c.s);
    }
}

```

Ans. Class A

Class B

Class C

```

(c).
public class classCommLine {
    public static void main(String args[]) {
        for(int i=0; i<args.length; i++)
            System.out.println("args[" + i + "]: " + args[i]);
    }
}

```

```

Ans. javac CommLine.java
java CommLine 45 65 75 85 33 44
args[0]: 45
args[1]: 65
args[2]: 75
args[3]: 85
args[4]: 33
args[5]: 44

```

Ques 2. Can abstract class have constructors in Java?

Ans 2. Yes abstract class can have constructors and you can call these constructors by using super keyword in derived class .

Ques 3. Create an abstract class 'Parent' with a method 'message'. It has two subclasses each having a method with the same name 'message' that prints "This is first subclass" and "This is second subclass" respectively. Call the methods 'message' by creating an object for each subclass.

Ans 3.

```
abstract class parent{
    abstract void message();
}

class first extends parent{
    @Override
    void message() {
        System.out.println("This is first subclass");
    }
}

class second extends parent{
    @Override
    void message() {
        System.out.println("This is second subclass");
    }
}

public class Main {

    public static void main(String[] args) {
        first f = new first();
        second s = new second();

        f.message();
        s.message();
    }
}
```

Ouput –

This is first subclass

This is second subclass

Ques 4. An abstract class has a construtor which prints "This is constructor of abstract class", an abstract method named 'a_method' and a non-abstract method which prints "This is a normal method of abstract class". A class 'SubClass' inherits the abstract class and has a method named 'a_method' which prints "This is abstract method". Now create an object of 'SubClass' and call the abstract method and the non-abstract method. (Analyse the result)

Ans 4.

```
abstract class parent{
    parent() {
        System.out.println("This is constructor of abstract class");
    }
}
```

```

    }

    abstract void a_method();

    void n_method(){
        System.out.println("This is normal method of abstract class");
    }
}

class subClass extends parent{
    @Override
    void a_method() {
        System.out.println("This is abstract method");
    }
}

public class Main {

    public static void main(String[] args) {
        subClass s = new subClass();
        s.a_method();
        s.n_method();
    }
}

```

the output is –

This is constructor of abstract class

This is abstract method

This is normal method of abstract class

Ques 5. Write a java code to find whether a number is prime or not where number is accepted from command line.

Ans 5.

```

public class prime
{
    public static void main(String args[])
    {
        int n=Integer.parseInt(args[0]);
        int cnt=0;
        for(int i=2;i<=n/2;i++)
        {
            if(n%i==0)
            {
                cnt=1;
            }
        }
        if(cnt==0)
    }
}

```

```
    {  
        System.out.println("no is prime"+n);  
    }  
    else  
    {  
        System.out.println("no is not prime"+n);  
    }  
}  
}
```

Output –

javac prime.java

java prime 92

92 is not prime.