

Cs_74 Introduction to internet programming

Question 1

Ans

Wrapper classes are classes that are used to make primitive variables into objects, and to make wrapped objects into primitives. int, boolean, double are all primitive data types and their respective wrapper classes are Integer, Boolean, and Double. Wrapper classes are useful in storing primitive data types in higher level data structures such as Stack<Object>, List<Object>, Queue<Object>.

Inner Classes In Java it is possible to define one class *inside* another. A class defined inside another one is called an *inner class*. Java provides two kinds of inner classes--static and non-static.

Wrapper classes

```
import java.util.*;
public class VecNum{
    public static void main(String argv[]){
        Vector v = new Vector();
        v.add(new Integer(1));
        v.add(new Integer(2));
        for(int i=0; i < v.size();i++){
            Integer iw =(Integer) v.get(i);
            System.out.println(iw.intValue());
        }
    }
}
```

Static Inner Classes

Consider the following Java code fragment:

```
public class A
{
    int y;

    public static class B
    {
        int x;

        void f () {}
    }
}
```

Non-Static Inner Classes

By default, an inner class is *non-static*:

```

public class A
{
    int y;

    public class B
    {
        int x;

        void f () {}
    }
}

```

Question 2

Ans

Importing a Class from an Existing Java Package

To import a Java Class, for example **v2k.awt.RGBColorSelector**, from an existing Java Package simply add either of the following statements to the top of your Java source file:

```

import v2k.awt.RGBColorSelector; // imports the RGBColorSelector
class only

```

or

```

import v2k.awt.*; // imports all classes in awt
package

```

If you use "**import v2k.awt.***", ALL classes in the awt package will be imported. We recommend importing only the classes you need. Bringing in an entire Package does not alter the size of the class file, but does lengthen the compile time. Wildcarding also increases the chances of namespace conflict as new classes are added to the package over time.

Question 3

Ans

```

import java.io.*;

```

```

class student
{
String s_name,s_cour;
int s_fee;
public
void insert(String nm,String co,int f)
{
s_name=nm;
s_cour=co;
s_fee=f;
}
}

```

```
//*****
```

```
class emp  
{
```

```
String e_name;  
int e_id_no,e_sal;
```

```
public
```

```
void insert(String nm,int id,int sl)  
{  
e_name=nm;  
e_id_no=id;  
e_sal=sl;  
}
```

```
void display()  
{
```

```
System.out.print(e_name +e_id_no +e_sal);  
}  
}
```

```
//*****
```

```
class mag extends student  
{
```

```
String m_name;  
int ac_no;
```

```
public
```

```
void m_insert()  
{
```

```
m_name="baluja Labs"  
ac_no=ac4736;  
}
```

```
void display_st()  
{
```

```
System.out.print(m_name +ac_no);
```

```

System.out.println(s_name +s_cour +s_fee);
}
}

//*****

class exe
{
public static void main(String arg[])throws IOException
{

BufferedReader br=new BUfferedReader(new InputStreamReader(System.in));

mag obj=new mag();

emp obj1=new emp();

String nm,co;
int f,id,s;

System.out.println("Enter student_name:--");
nm=br.readLine();

System.out.println("Enter course_name:--");
co=br.readLine();

System.out.println("Enter fee:--");
f=br.readLine();

//*****
System.out.println("Enter emp_name:--");
nm=br.readLine();

System.out.println("Enter id:--");
id=br.readLine();

System.out.println("Enter salary:--");
s=br.readLine();

obj.insert(nm,id,f);

obj1.insert();
obj1.display();
}
}

```

```
obj.m_insert();
obj.display_st();
}
```

```
}
```

Question 4

Ans

```
import java.io.*;
```

```
class Gr
{
public static void main(String arg[])throws IOException
{
int n1,n2;
```

```
BufferedReader br=new BufferedReader (new InputStreamReader(System.in));
```

```
System.out.println("Enter two number:---");
n1 = Integer.parseInt(br.readLine());
```

```
n2 = Integer.parseInt(br.readLine());
```

```
int gcd = 1;
int k = 1;
```

```
while (k <= n1 && k <= n2)
{
if (n1 % k == 0 && n2 % k == 0)
gcd = k;
k++;
}
```

```
System.out.println(gcd);
```

```
}
}
```

Question 5

Ans

```

import java.io.*;

class subject
{
public static void main(String arg[])throws IOException
{

int a,b,c;
BufferedReader br =new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter Marks of Maths:--");
a=Integer.parseInt(br.readLine());

System.out.println("Enter Marks of Physics:--");
b=Integer.parseInt(br.readLine());

System.out.println("Enter Marks of Chemistry:--");
c=Integer.parseInt(br.readLine());

if(a>=75 && b>=60 && c>=70)
System.out.println("Eligible candidates");
else
System.out.println("not eligible candidates");
}
}

```

Question 6

Ans

```

import java.io.*;

class comb
{
public static void main(String arg[])throws IOException
{

int i,j,k,k1;
int n[]=new int[4];

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter number:---");

for(i=0;i<4;i++)
n[i]=Integer.parseInt(br.readLine());

```

```

for(i=0;i<4;i++)
{
    for(j=0;j<4;j++)
    {
        for(k=0;k<4;k++)
        {
            for(k1=0;k1<4;k1++)
            {
                if(i!=j && i!=k && i!=k1 && j!=k && j!=k1 && k!=k1 )
                {
                    System.out.print(n[i]);
                    System.out.print(n[j]);
                    System.out.print(n[k]);
                    System.out.print(n[k1]);

                }

                System.out.println("");
            }
        }
    }
}

}
}
}
}

```

Question 7

Ans

Write a java program to count 1 to 1000 number using 10 threads?

```

class Number
{

private int ctr = 0;

public
void cnumber()
{
ctr = ctr + 1;
}
void print()

```

```
{  
System.out.println("NO :" +ctr);  
}
```

```
}
```

```
class Num extends Thread
```

```
{  
Number NN;
```

```
Num (Number NN)
```

```
{  
this.NN = NN();  
}
```

```
public void run()
```

```
{
```

```
for (int i = 0; i<1000; i++)
```

```
Number.cnumber();
```

```
}
```

```
}
```

```
class ravi
```

```
{
```

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

```
{
```

```
Number NN= new Number();
```

```
NN.start();
```

```
NN.print();
```

```
}
```

```
}
```