Accenture Technical & Coding

1. Can you differentiate between “var++” and “++var”?

Ans. Expressions “var++” and “++var” are used for incrementing the value of the “var” variable.

“var++” will first give the evaluation of expression and then its value will be incremented by 1, thus it is called as post-incrementation of a variable. “++var” will increment the value of the variable by one and then the evaluation of the expression will take place, thus it is called pre-incrementation of a variable.

Example:

```
/* C program to demonstrate the difference between var++ and ++var */

#include<stdio.h>

int main()

{

int x,y;

x=7, y=1;

printf(“%d %d”, x++, x); //will generate 7, 8 as output

printf(“%d %d”, ++y, y); //will generate 8, 8 as output

}
```

2.What is meant by the Friend function in C++?

Ans. A friend() function is a function that has access to private and protected members of another class i.e., a class in which it is declared as a friend. It is possible to declare a function as a friend function with the help of the friend keyword.

Syntax:

```
class class_name

{

//Statements

friend return_type function_name();

}
```

3.What is a map() function in Python?

Ans. In Python, the map() function is useful for applying the given function on every element of a specified iterable(list, tuple, etc.).

Syntax for map() function is: map(func, it)

Where func is a function applied to every element of an iterable and itr is iterable which is to be mapped. An object list will be returned as a result of map() function execution.

Example:

```
def addition(n):
return n+n
number=(10, 20, 30, 40)
res= map(addition, number)
print(list(res))
```

4. Write a C++ program for generating the Fibonacci series.

Ans.The Fibonacci series is a number sequence in which each number is the sum of the previous two numbers. Fibonacci series will have 0 followed by 1 as its first two numbers.

The Fibonacci series is as follows:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55….

The below-given program will display the Fibonacci series of n range of numbers given by the user. If the entered range value is 1, the num1 value will be printed, i.e., 0. If the entered range value is 2, num1 and num2 values will be printed, i.e., 0 and 1. If entered range value is n, num1 and num2 value will be printed. Along with that, each next term will be calculated based on the addition of the previous two numbers and this process continues until it generates n numbers in a Fibonacci series.

```
#include<iostream.h>
#include<conio.h>
void main()
{
int num1,num2,nextnum,n,i;
cout<<“Enter the value for range:”; //Fibonacci series range value will be inputted
cin>>n;
num1=0;
num2=1;
cout<<“Fibonacci series is:”<<endl;
if(n==1)
cout<<num1<<endl; //Single value will be printed if range value is 1
else if(n==2)
cout<<num1<<“\t”<<num2<<endl; //Two values will be printed if the range value is two
else
{
cout<<num1<<“\t”<<num2<<“\t”;
for(i=3;i<=n;i++) //Fibonacci series will be printed based on range limit
{
nextnum=num1+num2;
cout<<nextnum<<“\t”;
num1=num2;
num2=nextnum;
}
}
getch();
}
```

5.What is “Collection Framework” in Java?

Ans. Collection Framework in Java is an architecture for storing the classes, and interfaces and manipulating the data in the form of objects. There are two main interfaces in Collection Framework that are:

Java.util.Collection

Java.util.Map

6.What do you understand by runtime polymorphism?

Ans. Polymorphism is a method of performing “a single task in different ways.” Polymorphism is of two types

Runtime Polymorphism

Compile-time polymorphism

Here we will discuss runtime polymorphism.

Runtime Polymorphism- We can achieve runtime Polymorphism by method overriding in Java. And method overriding is a process of overriding a method in the subclass which is having the same signature as that of in superclass.

```
class A{ //Superclass
void name()
{
System.out.println(“this is student of Superclass”);
}
}
class Student extends A //Subclass
{
void name(){ // method Override with same signature(runtime polymorphism)
System.out.println(“this is student of subclass”);
}
public static void main (String[] args) {
A a= new A(); // refrence of A class
A b= new Student(); // refrence of student class
a.name();
b.name();
}
}
```

Output:

```
this is student of Superclass
this is student of subclass
```

7.What is List interface in collections?

Ans. List interface is an interface in Java Collection Framework. List interface extends the Collection interface.

It is an ordered collection of objects.

It contains duplicate elements.

It also allows random access of elements.

Syntax:

```
public interface List<E> extends Collection<E>
```

8.Can we implement multiple interfaces in a single Java class?

Ans. Yes, it is allowed to implement multiple interfaces in a single class. In Java, multiple inheritances is achieved by implementing multiple interfaces into the class. While implementing, each interface name is separated by using a comma(,) operator.

Syntax:

```
public class ClassName implements Interface1, Interface2,…, InterfaceN

{

//Code

}
```

9.What is an abstract class in Java?

Ans.An Abstract class is used to achieve abstraction in Java. If we use the keyword “abstract” with the class name, then it is called as an abstract class.

An Abstract class can have only methods without body or can have methods with some implementation.

The Abstract class cannot be instantiated

It’s not necessary that an abstract class should have an abstract method.

Syntax:

```
abstract class Student{

}
```

10.What do you understand by object cloning?

Ans. Object cloning is a mechanism of creating the same copy of an object. For object cloning, we can use clone() method of the Object class. The class must implement the java.lang.Clonable interface, whose clone we want to create otherwise it will throw an exception.

Syntax of clone() method:

```
protected Object clone() throws CloneNotSupportedException
```