Assignment-2
Part -A

Que-1)What are classes and objects in Java?
Class - class is user-defined datatype that define properties and behavious of object.
class contains variable, constructor and method
class keyword is used to declare class type
class doesnt occupy memory.

Object - Object is real-world entity that has state and behavious.
for example - Mobile is entity where color, price, model are state and switch on, swithc off, take photo are behaviours.
in java, object is class type memory location and new keyword is used to create object.

Que-2)What is the difference between primitive and reference data types?
primitive data types - 1)it store actual values directly.
2)primitive datatypes are int, float, double, char, boolean, byte, short, long.
3)it stores value in stack memory
4)it Passed by value to methods.

reference data types - 1)it store address of object.
2)reference datatypes are String, Arrays, Classes, Interfaces, Objects.
3)it stores in stack memory
4)it Passed by reference to methods.

Que-3)What are access modifiers in Java?
Access modifiers control who can access class, variable, constructor and method.
there are four types of access modifiers such as public, private, default, protected.

public - public access modifier can be access everywhere in program.
public can be applicable for class, variable, method, constructor.

private - private access modifiers can be access within class.
private can be applicable for variable, method, constructor, method and inner-class.

protected - protected can be access within class and its child classes.
protected can be applicable for variable, method, constructor, method and inner-class.

default - default access modifiers can be access within same package.
default can be applicable for class, variable, method, constructor, method and inner-class.

Que-4)Explain the concept of encapsulation.
Encapsulation means binding all data together and protecting it from unauthorized access. Just like a class binds variables, constructors, and methods together, encapsulation makes variables private and provides access only to authorized users through public getter and setter methods.

Que-5)What is inheritance and why is it used?

inheritance - when a class acquire properties and behavious of another class and make is-a ralationship that is called inheritance.

Extend keyword is used for inheritance.

inheritance provides code reusebility where child can resue exists functionality of parent-class.

types of inheritance-

1)single inheritance -when a class inherit only single parent class that is single inheritance.

2)multi-level inheritance - when a class inherit another class which has parent class that is multi-level inheritance.

3)Hierarchical Inheritance - whena Multiple child classes inherit from the same parent class that is hierarchical inheritance .

4)Multiple Inheritance - when a class inherits from multiple classes that is multiple inheritance.

java doesnt support mutliple inheritace.

why use inheritance - Inheritance allows a new class to reuse the properties and methods of an existing class.

You don't need to write the same code again.without inheritance, new class have to write same code.

parent class can hold child class object, it provides flexiblity in program

Part -B

Que-1) Swap two Number without third variable

```
import java.util.Scanner;

public class SwapTwoNum {

	public static void main(String[] args) {

		Scanner sc = new Scanner(System.in);
		System.out.println("Enter a : ");
		int a = sc.nextInt();
		System.out.println("Enter b : ");
		int b = sc.nextInt();

		a = a ^ b;
		b = a ^ b;
		a = a ^ b;

		System.out.println("After Swapping : \n" + a + "   " + b);

	}
}
```

Output -

```
Enter a :
12
Enter b :
31
After Swapping :
31   12
```

Que-2) Write a program to find the factorial of a number using loop.

```
import java.util.Scanner;

public class FactorialOfNum {

	public static void main(String[] args) {

		Scanner scanner = new Scanner(System.in);

		while (true) {
			System.out.print("Enter a N: ");
			int n = scanner.nextInt();

			if (n < 0) {
				System.out.println("Invalid !!");
			} else {
				long factorial = 1;
				for (int i = 1; i <= n; i++) {
					factorial *= i;
				}
				System.out.println("The factorial of " + n + " is:
" + factorial);
				break;
			}
		}

	}
}
```

```
Output -
Enter a N: 12
The factorial of 12 is: 479001600
```

Que-3)Write a program to print Fibonacci series up to n terms.

```
import java.util.Scanner;

public class FibonacciSeries {

	public static void main(String[] args) {

		Scanner sc = new Scanner(System.in);
		int n;
		while (true) {

			System.out.println("Enter a N: ");
			n = sc.nextInt();

			if (n <= 0) {
				System.out.println("Please enter a positive
integer.");
			} else {
				int a = 0, b = 1;
				System.out.print("Fibonacci Series up to " + n + "
terms: ");
				for (int i = 1; i <= n; i++) {
					System.out.print(a + " ");
					int next = a + b;
					a = b;
					b = next;
```

```
                        }
                        break;
                  }
            }

      }
}
```

Output -

```
Enter a N:
12
Fibonacci Series up to 12 terms: 0 1 1 2 3 5 8 13 21 34 55 89
```

Que-4)Write a program to reverse a number.

```
import java.util.Scanner;

public class ReverseNumber {

      public static void main(String[] args) {

            Scanner sc = new Scanner(System.in);

            System.out.println("Enter a N: ");
            int n = sc.nextInt();
            int rev = 0;
            while (n != 0) {
                  int digit = n % 10;
                  rev = rev * 10 + digit;
                  n /= 10;
            }
            System.out.println("Reversed Number: " + rev);

      }
}
```

Output -

```
Enter a N:
132
Reversed Number: 231
```

Que-5)Write a program to check if a number is palindrome.

```
import java.util.Scanner;

public class IsPalindrome {

      public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter a N:");
            int n = sc.nextInt();
            int temp = n;
            int rev = 0;
            while (n > 0) {
                  int rem = n % 10;
                  rev = rev * 10 + rem;
                  n /= 10;
            }

            if (temp == rev)
```

```
                System.out.println("Given Number is Palindrome !!");
          else
                System.out.println("Not Palindrome !!");

     }
}
```

Output -

```
Enter a N:
121
Given Number is Palindrome !!
```

GitHub Link -
https://github.com/YogeshPathade01/Java_Coding_Questtions.git