**Core Java Assignment 3 (string)**

**Question 1**

public class length {

    public static void main(String[] args) {

        String str = "Hello World";

        int count = str.length();

        System.out.println("The string has "+count+"characters");

    }

}

Output

The string has 11characters

Question 2

public class Concat {

    public static void main(String[] args) {

        String s = "Hello,"+"How are you?";

        System.out.println(s);

    }

}

Output

Hello,How are you?

Question 3

A

public class LowerCase {

    public static void main(String[] args) {

        String s1 = "Java String pool refers to collection of Strings which are stored in heap memory";

        String s1lower=s1.toLowerCase();

        System.out.println(s1lower);

    }

}

Output- java string pool refers to collection of strings which are stored in heap memory

B

public class UpperCase {

    public static void main(String[] args) {

        String s1 = "Java String pool refers to collection of Strings which are stored in heap memory";

        String s1upper=s1.toUpperCase();

        System.out.println(s1upper);

    }

}

Output- JAVA STRING POOL REFERS TO COLLECTION OF STRINGS WHICH ARE STORED IN HEAP MEMORY

C

public class replace {

    public static void main(String[] args) {

        String s1="Java String pool refers to collection of Strings which are stored in heap memory";

        String replaceString=s1.replace('a','$');

        System.out.println(replaceString);

    }

}

Output- J$v$ String pool refers to collection of Strings which $re stored in he$p memory

D

public class Contains {

    public static void main(String[] args) {

        String name= "Java String pool refers to collection of Strings which are stored in heap memory";

        System.out.println(name.contains("collection"));

    }

}

Output- true

E

public class Contains {

    public static void main(String[] args) {

        String name= "Java String pool refers to collection of Strings which are stored in heap memory";

        System.out.println(name.contains("Java String pool refers to collection of Strings which are stored in heap memory"));

    }

}

Output- true

**Assignments on StringBuffer**

Question 1

class StringBufferExample{

    public static void main(String args[]){

    StringBuffer sb=new StringBuffer("StringBuffer");

    sb.append("is a peer class of String" + " that provides much of "+ " the functionality of strings");

    System.out.println(sb);

    }

    }

Output- StringBuffer is a peer class of String that provides much of the functionality of strings

Question 2

class StringBufferExample2{

    public static void main(String args[]){

    StringBuffer sb=new StringBuffer("It is used to at the specified index position");

    sb.insert(14 ,"insert text");

    System.out.println(sb);

    }

    }

Output- It is used to insert text at the specified index position

Question 3

class StringBuffer3 {

    public static void main(String args[]){

        StringBuffer sb=new StringBuffer("This method returns the reversed object on which it was called");

        sb.reverse();

        System.out.println(sb);

    }

}

Output- dellac saw ti hcihw no tcejbo desrever eht snruter dohtem sihT

Assignments on StringBuilder class

Question 1

class StringBuilderExample{

    public static void main(String args[]){

    StringBuilder sb=new StringBuilder("StringBuilder");

    sb.append(" is a peer class of String" + " that provides much of "+ " the functionality of strings");

    System.out.println(sb);

    }

    }

Output- StringBuilder is a peer class of String that provides much of the functionality of strings

Question 2

class StringBuilder2 {

    public static void main(String args[]){

        StringBuilder sb=new StringBuilder("It is  used to at the specified index position");

        sb.insert(14 ,"insert text");

        System.out.println(sb);

        }

        }

Output- It is used to insert text at the specified index position

Question 3

class StringBuilder3 {

    public static void main(String args[]){

        StringBuilder sb=new StringBuilder("This method returns the reversed object on which it was called");

        sb.reverse();

        System.out.println(sb);

    }

}

Output- dellac saw ti hcihw no tcejbo desrever eht snruter dohtem sihT