**----------------------------------------------------------------------------------------------**

Date: 01/03/2022

**Homework Question:-**

**Q1.For soccer player class write down the data member and member function.**

Class🡪 Soccer

Data member🡪 Player Name, Player number, point scored

Member function🡪getPlayerData(), displayPlayerData(), getScored(),

displayScored()

**Q2. Write and flow chart to determine a student grade and indicate whether its failing or passing. The final grade is calculated as 4 marks physics, biology, chemistry, maths.**

Solution:- step1: start the program

Step2: Enter student name and roll number

Step3: input marks physics, biology ,chemistry , maths

Var ph,bio,chem,math

Step4: var total

Total=ph+bio+chem+math

Step5: var average

Average=total/4

Step6: If average (avg>40) = passed

Else failed Grade C

Step7: if average (avg>=60) = firstclass Grade A

Else if (avg>40 && avg<60) second class Grade B

Step8:End

**Q3 Write and algorithms or flow chart for getting married.**

Solution: Step1: start the program

Step2:Enter the age and Gender

Var age , Gender

Step3:Enter marrital status

Var marritalstatus

Step4: if maritalstatus==m

“You can not marry”

Step5: else if marritalstatus==u

If gender==m

Age>21

“You can marry”

Else if“you can not marry”

Else if gender==w

Age>18

“You can marry”

Else You can not marry

Step6: else

“Invalid input”

Step7: End

Q4 **Write and algorithms or flow chart to post a letter.**

Solution: step1: start the program

Step2: write the letter

Step3:write down the receiver address

Step4: Put it in envelope

Step5: paste a stamp sticker

Step6:put it in a letter box

Step7: End

**Q5. Write and algorithms or flow chart to catch a school bus.**

Solution: step1: start the program

Step2: wake up at early

Step3: get ready for the school

Step4: go to school bus stop

Step5: catch the school bus

Step6: End

**----------------------------------------------------------------------------------------------**

**Assignment 1**

**----------------------------------------------------------------------------------------------**

**Write Algorithm & Flowchart for the following programs**

**1)Check the given number is EVEN or ODD?**

Step 1: Start.

Step 2: Read a number from user to N.

Step 3: Divide the number by 2 and store the remainder in R.

Step 4: If R = O Then go to Step 6.

else

Step 5: Print “N is odd” go to step 7.

Step 6: Print “N is even”

Step 7: Stop.

**2) Write a Java Program to find the Factorial of given number?**

Step 1: Start

Step 2: Declare Variable n, fact, i

Step 3: Read number from User

Step 4: Initialize Variable fact=1 and i=1

Step 5: Repeat Until i<=number

5.1 fact=fact\*i

5.2 i=i+1

Step 6: Print fact

Step 7: Stop

We first take input from user and store that value in variable named “n”.

Then we initialize a variable “Fact” with value 1 (i.e Fact=1) and variable i with value 1(i.e i=1).

Repeat next two steps until i is less than n.

Multiply Fact with current value of i

Increment i with 1

At last, print the value of Fact.

Let’s take an example,

Let the input be 4.

The equation that gets created by our algorithm is 4x3x2x1.

So, The Factorial of 4 is 24(4x3x2x1).

**3) Find the Factorial of a number using Recursion?**

step 1: start.

step 2: give input from user n (give no from user).

step 3: give function call

factorial(n).(goto the step 6)

step 4: print output;

step 5: stop;

step 6: give a condition if(n==1)

return 1.

else

again call function until( n==1)

factorial(n).

**4)Swap two numbers without using third variable approach?**

STEP 1: START.

STEP 2: take a input from user x, y.

STEP 3: print the given input.

STEP 4: adding given numbers and store it into any one variable out of two

x = x + y.

STEP 5: y= x - y.

STEP 6: x =x - y.

STEP 7: PRINT x, y.

STEP 8: END.

EX. suppose to be the given no be

X=10 ,Y=15.

adding X and Y store it into X.

X=10+15=25.

subtraction will be happen

Y=X-Y=25-15=10

y=10;===========>number will be swap.

X=X-Y=25-10=15

X=15============>number will be swap.

**5) How to check the given number is Positive or Negative in Java?**

STEP 1: Start.

STEP 1: Declare a variable.

STEP 2: Initialize the variable.

STEP 3: Call a function to check whether the number is positive or negative.

STEP 4: there is also use one loop

4.1)if the result is true it print no is positive

4.2)if the result is false it print no is negative

STEP 5: Stop.

**6) Write a Java Program to find whether given number is Leap year or NOT?**

STEP 1: start.

STEP 2: declare a one variable i.e.X;

STEP 3: take a input from user and store it into declared variable.

STEP 4:by using nested if else condition;

4.1) if (X%400==0||X%2==0)

{

print given input year is leap year

}

else

{

print given year is not a leap year

}

STEP 5: stop.

**7)Write a Java Program to Print 1 To 10 Without Using Loop ?**

STEP 1: start;

STEP 2: declare one variable and initialize it with 1;

STEP 3:calling the recursive function...function name will be XYZ passing

the argument as given no.

STEP 4: logic should be in function we use recursion

XYZ(A)

{

if(x==11)

return 1;

else

print the given no.

XYZ(A++)=============>passing argument will be incremented A.

}

STEP 5:stop.

**8)Write a Java Program to print the digits of a Given Number?**

STEP 1: start;

STEP 2: give input from user as number n , another variable num.

STEP 3:while(n!=0)

{

num=n%10;

n=n/10;

print the num;

}

step 4: stop.

**9)Write a Java Program to print all the Factors of the Given number?**

STEP 1: start the program.

STEP 2: take a input from user as num.

STEP 3: appalying a loop.

for(i=1;i<=num;i++)

{

if(num%i==0)

print given i is the factorial of num.

else

print given i is not a factor of num.

}

STEP 4:stop

**10)Write a Java Program to find sum of the digits of a given number?**

STEP 1: start;

STEP 2: give input from user as number n , another variable num.

STEP 3: take one variable name as a sum initialize as a 0;

STEP 4:while(n!=0)

{

num=n%10;

sum=sum+num;

n=n/10;

}

STEP 5: print the sum .

STEP 6:stop.

**11)Write a Java algorithm to find the smallest of 3 numbers (a,b,c)?**

STEP 1: start the programm.

STEP 2: declare the three variables as a,b,c;

STEP 3: takes input from user.

STEP 4: applying nested loop in that condition,

step 5: if(a<b)

{

if(b<c)

number a is smallest number.

else

no a is not a smallest number

}

if(b<c)

{

if(c<a)

number b is smallest number.

else

no b is not a smallest number

}

if(c<a)

{

if(a<b)

number c is smallest number.

else

no c is not a smallest number

}

STEP 6: stop.

**13) Write a java program to Reverse a given number?**

STEP 1: start the given program.

STEP 2: enter the four digit num from user.

STEP 3: declare the variable num1,num2,num3,num4.

STEP 4: num1=num%10;

num2=num1%10;

num3=num2%10;

num4=num3%10;

STEP 5:print the given number in reverse order

print num1,num2,num3,num4

step 6: stop.

**14)Write a Java Program to find GCD of two given numbers?**

STEP 1: start the programm

STEP 2: take two number num1,num2 from user, declare the one variable as i.

STEP 3: applay the loop

STEP 4:for(i=1;num1%i==0 && num2%i==0;i++);

STEP 5: print the i,GCD of given two number is i;

STEP 6: stop

**15) Write a java program to LCM of TWO given number?**

STEP 1: start the programm

STEP 2: take two number num1,num2 from user, declare the one variable as i.

STEP 3: applay the loop

STEP 4:for(i=1;num1%i==0 || num2%i==0;i++);

STEP 5: print the i,LCM of given two number is i;

STEP 6: stop

**17)Check whether the Given Number is a Palindrome or NOT?**

STEP 1: start the given program.

STEP 2: enter the num from user.

STEP 3: declare the variable num,reminder,reverse=0;

STEP 4: declare variable n=num assign given number in declare variable;

STEP 4: while(n!=0)

{

reminder=num%10;

reverse=reverse\*10+reminder;

num=num/10;

}

STEP 5:if(reverse==num)

print given number as palindeome.

else

print given number is not palindrome

step 6: stop.

**18)Write a Java Program to print all the Prime Factors of the Given Number?**

STEP 1:start

STEP 2:take number as input from user.

STEP 3:appalying loop

for(i=2;i<num;i++)

{

while(num%i==0)

{

print the i;

}

if(num>2)

print the num.

}

STEp 4: stop the program.

**19)To print the following series EVEN number Series 2 4 6 8 10 12 14 16 .....?**

STEP 1: start the program

STEP 2: gives a input from user as num.

STEP 3:initilize the variablen i=1.

STEP 4:appalying a loop

step 5:

for(i;i<=num;i++)

{

if(n%2==0)

print the given num is even.

print i.

}

STEP 6:stop

**20)To print the following series ODD number Series 1 3 5 7 9 11 13 ....?**

STEP 1: start the program

STEP 2: gives input from user as num

STEP 3:initilize the variablen i=1.

STEP 4:appalying a loop

step 5:

for(i;i<=num;i++)

{

if(n%2!=0)

print the given number is odd

print the i.

}

STEP 6:stop

**--------------------------------------------------------------------------------------------------------**

**Date: 02/03/2022**

**Homework**

**Q1.Write down the keyword in java language.**

|  |  |  |
| --- | --- | --- |
| S.No | Keyword | Usage |
| 1. | **abstract** | Specifies that a class or method will be implemented later, in a subclass |
| 2. | **assert** | Assert describes a predicate placed in a java program to indicate that the developer thinks that the predicate is always true at that place. |
| 3. | **boolean** | A data type that can hold True and False values only |
| 4. | **break** | A control statement for breaking out of loops. |
| 5. | **byte** | A data type that can hold 8-bit data values |
| 6. | **case** | Used in switch statements to mark blocks of text |
| 7. | **catch** | Catches exceptions generated by try statements |
| 8. | **char** | A data type that can hold unsigned 16-bit Unicode characters |
| 9. | **class** | Declares a new class |
| 10. | **continue** | Sends control back outside a loop |
| 11. | **default** | Specifies the default block of code in a switch statement |
| 12. | **do** | Starts a do-while loop |
| 13. | **double** | A data type that can hold 64-bit floating-point numbers |
| 14. | **else** | Indicates alternative branches in an if statement |
| 15. | **enum** | A Java keyword is used to declare an enumerated type. Enumerations extend the base class. |
| 16. | **extends** | Indicates that a class is derived from another class or interface |
| 17. | **final** | Indicates that a variable holds a constant value or that a method will not be overridden |
| 18. | **finally** | Indicates a block of code in a try-catch structure that will always be executed |
| 19. | **float** | A data type that holds a 32-bit floating-point number |
| 20. | **for** | Used to start a for loop |
| 21. | **if** | Tests a true/false expression and branches accordingly |
| 22. | **implements** | Specifies that a class implements an interface |
| 23. | **import** | References other classes |
| 24. | **instanceof** | Indicates whether an object is an instance of a specific class or implements an interface |
| 25. | **int** | A data type that can hold a 32-bit signed integer |
| 26. | **interface** | Declares an interface |
| 27. | **long** | A data type that holds a 64-bit integer |
| 28. | **native** | Specifies that a method is implemented with native (platform-specific) code |
| 29. | **new** | Creates new objects |
| 30. | **null** | This indicates that a reference does not refer to anything |
| 31. | **package** | Declares a Java package |
| 32. | **private** | An access specifier indicating that a method or variable may be accessed only in the class it’s declared in |
| 33. | **protected** | An access specifier indicating that a method or variable may only be accessed in the class it’s declared in (or a subclass of the class it’s declared in or other classes in the same package) |
| 34. | **public** | An access specifier used for classes, interfaces, methods, and variables indicating that an item is accessible throughout the application (or where the class that defines it is accessible) |
| 35. | **return** | Sends control and possibly a return value back from a called method |
| 36. | **short** | A data type that can hold a 16-bit integer |
| 37 | **static** | Indicates that a variable or method is a class method (rather than being limited to one particular object) |
| 38. | **strictfp** | A Java keyword is used to restrict the precision and rounding of floating-point calculations to ensure portability. |
| 39. | **super** | Refers to a class’s base class (used in a method or class constructor) |
| 40. | **switch** | A statement that executes code based on a test value |
| 41. | **synchronized** | Specifies critical sections or methods in multithreaded code |
| 42. | **this** | Refers to the current object in a method or constructor |
| 43. | **throw** | Creates an exception |
| 44. | **throws** | Indicates what exceptions may be thrown by a method |
| 45. | **transient** | Specifies that a variable is not part of an object’s persistent state |
| 46. | **try** | Starts a block of code that will be tested for exceptions |
| 47. | **void** | Specifies that a method does not have a return value |
| 48. | **volatile** | This indicates that a variable may change asynchronously |
| 49. | **while** | Starts a while loop |

**Q2 Write down the summery of data type, size ,default value,range**