Set B

Scoring Indicators

COURSENAME: PROBLEM SOLVING AND PROGRAMMING

COURSECODE:2131

QID: 2106220110

QNo	Scoring Indicators	Splits core	SubTotal	Totals core
	PART A			**9**
I.1	x = x + 2	1	1	9
I.2	int, float, double, char	2x ¼	1	

I.3	if(test_expression)
	{
	statement 1;
	statement 2;
	}
		1	1	

I.4	False	1	1	
I.5	Logical AND operator	1	1	
I.6	return	1	1	
I.7	Scanf(), printf(), getch(), getchar() etc(any two)	1	1	
I.8	[0][0]	1	1	
I.9	**Definition - 1** An array is a group (or collection) of same data types stored at contiguous memory locations.	1	1	
	PART B			**24**
II.1	**Input – 1, Logic -1 , output - 1** Sample algorithm Step 1: Start. Step 2: Read amount. Step 3: Read years. Step 4: Read rate. Step 5: Calculate the interest with the formula "Interest=Amount*Years*Rate/100". Step 6: Print interest. Step 7: Stop.	1+1+1	3	3

II.2	**= operator – 1, == operator 1, example 1** = Assignment operator used to assign the result of an expression or value to a variable Eg: x = a + b // value of the expression is assigned to x = = Equal to operator is a relational operator and is used to check equality of two values. Eg: a = = b // (a is equal to b)	1+1+1	3	3

II. 3	**Correct Implementation - 3**
	
	int main()
	{
	int i = 0;
	while (i<10)
	{
	printf("\n%d", i);
	i++;
	}
	return 0;
	}
		3	3	3

II. 4	**Correct Syntax – 3**
	Sample Syntax
	switch(variable)
	{
	case 1:
	//execute your code
	break;
	case n:
	//execute your code
	break;
	default:
	//execute your code
	break;
	}
		3	3	3

II. 5	**Program structure – 1, Factorial calculation 2**
	
	Sample program
	#include <stdio.h>
	int main()
	{
	int i, Number;
	long Factorial = 1;
	printf("\n Enter any number to Find Factorial\n");
	scanf("%d", &Number);
	for (I = 1; I<= Number; i++)
	{
	Factorial = Factorial * I;
		3	3	3

	}
	printf("\nFactorial of %d = %d\n", Number, Factorial);
	return 0;
	}			

II. 6	Definition Syntax Example	1 1 1	3	3
II. 7	Correct Function definition	3	3	3
II. 8	**Declaration with initialization – 2, Example - 1** **Declaration of two dimensional Array** The syntax for declaring a Two-dimensional array is data_type array_name[rows][columns]; Eg: int A[10][5]; **Initializing two dimensional arrays** Two ways to initialize a two dimensional array during declaration. **Example :** int stud[4][2]= { 1234,56, 1212,33, 1434,80, 1312,78 }; int stud[4][2]={ { 1234, 56 }, { 1212, 33 }, { 1434, 80 }, { 1312, 78 }, }	2 + 1	3	3
II. 9	Correct segment	3	3	3
II. 10	Code segment to read elements	3	3	3
	PART C			**42**
III. 1	**Definition -1 , listing with explanation - 1 mark each** The program development life cycle is a set of steps or phases which are used to develop a program in any programming language. • Program development life cycle contains 6 phases, which are : • Problem Analysis • Problem Design – Algorithm, Flowchart and Pseudo code • Coding • Compilation and Execution • Debugging and Testing • Documentation	1 + 1*6	7	7
III. 2	**Definition – 1, Types – 1, Implicit -1, explicit - 1** (a) Typecasting is converting one data type into another one. Two types – Implicit and explicit conversion	1+1+1+1	4	7

	Implicit Conversion–C automatically converts any intermediate values to the proper type, so the expression can be evaluated without loosing any significance. If the operands are of different types, the *lower* type is automatically converted to the *higher* type before the operation proceeds. The result is of the higher type.
	Example :
	short a=10; //initializing variable of short data type
	int b; //declaring int variable
	b=a; //implicit type casting
	Explicit Conversion –It is user defined.The user can type cast the result to make it of a particular data type.
	Syntax : (type) expression, where, type is the standard 'C' data type and expression can be a constant, a variable or an actual expression.
	Example:
	float a = 1.2;
	//int b = a; //Compiler will throw an error for this
	int b = (int)a + 1;
	printf("Value of a is %f\n", a			

	(b)
	Variable declaration -1, input/output – 1, Logic – 1
	
	Sample Program
	#include <stdio.h>
	int main()
	{
	float Celcius, Farenheit;
	printf("Enter temperature in Celsius: ");
	scanf("%f", &Farenheit);
	Farenheit = (Celcius * 9 / 5) + 32;
	printf("%.2f Celsius = %.2f Fahrenheit", Celcius, Fahrenheit);
	return 0;
	}
		1+1+1	3	

III.3	(a) printf() – 1 scanf() – 1 Example – 2 **(b) Arithmetic Operators – 3, Relational operator - 2 , Logical operators – 2** Relational Operators - >, <, >=, <=, = =, != Logical Operators - &&, ||, ! Explanation	1+1+2 1+1+1	4 3	7
III. 4	(a) Syntax Explanation/Working Example	1 2 1	4	7

	(b)
	Declaration – 1, input/output – 1, Logic - 1
	
	Sample Program
	int main()
	{
	int I, Number;
	printf("\n Enter any number to Find Factors \n");
	scanf("%d", &Number);
	printf("\n Factors of the Given Number are:\n");
	for (I = 1; I<= Number; I++)
	{
	if(Number%I = = 0)
	{
	printf(" %d ", i);
	}
	}
	return 0;
	}
		Declaration – 1
		Input/output – 1
		Logic – 1
			3	

III. 5	**(a) Declaration – 1, input – 1, Logic – 2**
	switch (gp) {
	case 10: Grade = 'S'; break;
	case 9: Grade = 'A'; break;
	case 8: Grade = 'B'; break;
	case 7: Grade = 'C'; break;
	case 6: Grade = 'D'; break;
	case 5: Grade = 'E'; break;
	default: Grade = 'F';
	}
		1+1+2	4	7

	(b) Syntax – 2 Example - 1	2+1	3	

III. 6	**Declaration – & input -1, roots – 3,**
	Sample program
	int main()
	{
	float a, b, c, d, root1, root2, realPart, i magPart;
	printf("Enter coefficients a, b and c: ");
	scanf("%f %f %f", &a, &b, &c);
	d = b * b - 4 * a * c;
	// condition for real and different roots
	i f (d > 0)
	{
	root1 = (-b + sqrt(d)) / (2 * a);
	root2 = (-b - sqrt(d)) / (2 * a);
	printf(" The roots are real and distinct, the roots are : \n");
	printf("root 1 = %.2f and root2 = %.2f", root1, root2);
	}
	// condition for real and equal roots
	else if (d == 0)
	{
	root1 = root2 = -b / (2 * a);
	printf(" The roots are real and equal and the roots are : \n");
	printf("root1 = root2 = %.2f;", root1);
	}
	// if roots are not real
	else
	{
	realPart = -b / (2 * a);
	imagPart = sqrt(-d) / (2 * a);
	printf(" The roots are i maginary and the roots are : \n");
	printf("root1 = %.2lf+%.2lfi and root2 = %.2f-%.2fi",
	realPart, i magPart, realPart,imagPart);
	}
	return 0;
	}
		1 + 3	4	7

	(b)if – else ladder – 1 nested if – 1 Example – 1	1+1+1	3	
III. 7	User defined functions – 4 Built –in functions - 3	4 + 3	7	7

III. 8	**Function definition – 4, Main function – 2, Function call - 1**<u>Sample Program</u>
	#include<stdio.h>
	int fact(int n){
	int i,f=1;
	for(i=1;i<=n;i++) {
	f=f*i; }
	return f;
	}
	void main() {
	int n,r,ncr;
	printf("Enter the value of n : \n");
	scanf("%d",&n);
	printf("Enter the value of r : \n");
	scanf("%d",&r);
	ncr=fact(n)/(fact(r)*fact(n-r));
	printf("Value of %dC%d = %d\n",n,r,ncr);
	}
		Factorial fn – 4
		Main fn – 2
		Function call – 1
			7	

III. 9	Declaration – 1 Input – 1 Elements reading – 2 Sorting – 2 Output - 1	3 +1+1+2	7	7

III.10	**Array declaration and input – 2, Searching logic – 4, Output - 1**
	
	<u>Sample Program</u>
	#include<stdio.h>
	
	int main()
	{
	int arr[10], Size, i, Search, Flag;
	
	printf("\n Enter the size of an array : ");
	scanf("%d",&Size);
	
	printf("\n Enter %d elements of an array: \n", Size);
	for(i = 0; i < Size; i++)
	{
	scanf("%d",&arr[i]);
	}
	
	printf("\n Enter the Search Element : ");
	scanf("%d",&Search);
	
	Flag = 0;
		Array declaration and reading – 2
		Search element reading – 1
		Search logic – 4
			7	7

	for(i = 0; i < Size; i++)
	{
	if(arr[i] == Search)
	{
	Flag = 1;
	break;
	}
	}
	
	if(Flag == 1)
	{
	printf("\n Search Element %d at Position %d ", Search, i + 1);
	}
	else
	{
	printf("\n Not found the the Search Element %d ", Search);
	}
	return 0;
	}			

III. 11	Declaration – 1 Read Matrix elements – 2 Find transpose – 2 Display Matrix - 2	1+1+2+1 +2	7	7
III. 12	Matrix reading – 3 Sum of row elements – 3 Output - 1	3 +3+1	7	7