**Assignment on C++ Structure**

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

**1.**Give the output of the following program. Assuming all the desired header files are already included, which are required to run the code.

struct Pixel  
{  
            int C, R;  
};

void Display(Pixel P)  
{  
            cout << "Col "<< P.C << " Row " << P.R << endl;  
}

int main()  
{            Pixel X = {40,50}, Y, Z;  
            Z = X;  
            X.C += 10;  
            Y = Z;  
            Y.C += 10;  
            Y.R += 20;  
            Z.C -= 15;  
            Display(X);  
            Display(Y);  
            Display(Z);

            return 0;  
}

Give the answer.

**Col 50 Row 50**

**Col 50 Row 70**

**Col 25 Row 50**

|  |  |  |  |
| --- | --- | --- | --- |
| **Obj X** | **Obj Y** | **Obj Z** | **Main Function line no** |
| **C=40, R=50** | **-** | **-** | **1** |
| **C=40, R=50** | **-** | **C=40, R=50** | **2** |
| **C=50, R=50** | **-** | **C=40, R=50** | **3** |
| **C=50, R=50** | **C=40, R=50** | **C=40, R=50** | **4** |
| **C=50, R=50** | **C=50, R=50** | **C=40, R=50** | **5** |
| **C=50, R=50** | **C=50, R=70** | **C=40 R=50** | **6** |
| **C=50, R=50** | **C=50, R=70** | **C=25, R=50** | **7** |
| **Printing Col 50 Row 50** | **-** | **-** | **8** |
| **-** | **Printing Col 50 Row 70** | **-** | **9** |
| **-** | **-** | **Printing Col 25 Row 50** | **10** |

---------------------------------------------------------------------------------------------------------------------------

**2.**Find the output of the following program. Assuming all the desired header files are already included, which are required to run the code.

struct Play  
{  
            int score, bonus;  
};

void calculate(Play &P, int N = 10)  
{  
            P.score++;  
            P.bonus += N;  
}

int main()  
{  
            Play PL = {10, 15};  
            calculate(PL, 5);  
            cout << PL.score << ":" << PL.bonus << endl;  
            calculate(PL);  
            cout << PL.score << ":" << PL.bonus << endl;  
            calculate(PL, 15);  
            cout << PL.score << ":" << PL.bonus << endl;

            return 0;  
}

Give the answer.

**11:20**

**12:30**

**13:45**

|  |  |
| --- | --- |
| **Obj PL** | **Main function line no.** |
| **Score=10,bonus=15** | **1** |
| **Score=11,bonus=20** | **2** |
| **Display 11:20** | **3** |
| **Score=12,bonus=30** | **4** |
| **Display 12:30** | **5** |
| **Score=13,bonus=45** | **6** |
| **Display 13:45** | **7** |

---------------------------------------------------------------------------------------------------------------------------

**3.**Find the output of the following program. Assuming all the desired header files are already included, which are required to run the code.

struct MyBox  
{  
            int length, breadth, height;  
};

void dimension (MyBox M)  
{  
            cout << M.length << "x" << M.breadth << "x";  
            cout << M.height << endl;  
}

int main ()  
{  
            MyBox B1 = {10, 15, 5}, B2, B3;  
            ++B1.height;  
            dimension(B1);  
            B3 = B1;  
            ++B3.length;  
            B3.breadth++;  
            dimension(B3);  
            B2 = B3;  
            B2.height += 5;  
            B2.length--;  
            dimension(B2);

           return 0;  
}

Give the answer.

**Output:**

**10x15x6**

**11x16x6**

**10x16x11**

|  |  |  |  |
| --- | --- | --- | --- |
| **B1** | **B2** | **B3** | **Main function line no** |
| **B1 = {l=10, b=15,h= 5}** | **-** | **-** | **1** |
| **B1 = {10, 15, 6}** | **-** | **-** | **2** |
| **Prints 10x15x6** | **-** | **-** | **3** |
| **B1 = {10, 15, 6}** | **-** | **B3= {10, 15, 6}** | **4** |
| **B1 = {10, 15, 6}** | **-** | **B3 = {11, 15, 6}** | **5** |
| **B1 = {10, 15, 6}** | **-** | **B3 = {11, 16, 6}** | **6** |
| **B1 = {10, 15, 6}** |  | **Prints 11x16x6** | **7** |
| **B1 = {10, 15, 6}** | **B2 = {11, 16, 6}** | **B3 = {11, 16, 6}** | **8** |
| **B1 = {10, 15, 6}** | **B2 = {11, 16, 11}** | **B3 = {11, 16, 6}** | **9** |
| **B1 = {10, 15, 6}** | **B2 = {10, 16, 11}** | **B3 = {11, 16, 6}** | **10** |
| **B1 = {10, 15, 6}** | **Prints 10x16x11** | **B3 = {l=11, b=16,h= 6}** | **11** |

---------------------------------------------------------------------------------------------------------------------------

**4.**Rewrite the following program after removing the syntactical errors (if any). Underline each correction.  
  
struct Pixels  
{  
            int color, style;  
}

void showPoint(Pixels P)  
{  
            cout << P.color, P.style << endl;  
}

int main()  
{  
            Pixels Point1 = (5, 3);  
            showPoint(Point1);  
            Pixels Point2 = Point1;  
            color.Point1 += 2;  
            showPoint(Point2);

            return 0;  
}

Give the answer.

**struct Pixels  
{  
            int color, style;  
}**

**void showPoint(Pixels P)  
{  
            cout << P.color**<<**P.style << endl;  
}**

**int main()  
{  
            Pixels Point1 = {5, 3};  
            showPoint(Point1);  
            Pixels Point2 = Point1;  
            Point1.color += 2;  
            showPoint(Point2);**

**return 0;  
}**

**Output:**

**53**

**53**

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

**5.**Declare a structure to represent a complex number (a number having a real part and imaginary part). Write C++ functions to add, subtract, multiply and divide two complex numbers.

**Program:**

**#include<iostream>**

**using namespace std;**

**struct complex**

**{**

**float real;**

**float imag;**

**}c1,c2;**

**class ComplexNum{**

**float a,b;**

**public:**

**void Addition(){**

**a=(c1.real)+(c2.real);**

**b=(c1.imag)+(c2.imag);**

**cout<<"Addition:"<<a<<"+"<<b<<"i"<<endl;**

**}**

**void Subtraction(){**

**a=(c1.real)-(c2.real);**

**b=(c1.imag)-(c2.imag);**

**cout<<"Subtraction: "<<a<<"+"<<b<<"i"<<endl;**

**}**

**void Multiplication(){**

**a=((c1.real)\*(c2.real))-((c1.imag)\*(c2.imag));**

**b=((c1.real)\*(c2.imag))+((c2.real)\*(c1.imag));**

**cout<<"Multiplication: "<<a<<"+"<<b<<"i"<<endl;**

**}**

**void Division(){**

**a=(((c1.real)\*(c2.real))+((c1.imag)\*(c2.imag)))/(((c2.real)\*(c2.real))+((c2.imag)\*(c2.imag)));**

**b=((((c2.real)\*(c1.imag))-((c1.real)\*(c2.imag)))/(((c2.real)\*(c2.real))+((c2.imag)\*(c2.imag))));**

**cout<<"Division: "<<a<<"+"<<b<<"i"<<endl;**

**}**

**};**

**int main(){**

**ComplexNum cm;**

**cout<<"Enter real and imaginary part of 1st complex number:";**

**cin>>c1.real>>c1.imag;**

**cout<<"Enter real and imaginary part of 2nd complex number:";**

**cin>>c2.real>>c2.imag;**

**cm.Addition();**

**cm.Subtraction();**

**cm.Multiplication();**

**cm.Division();**

**return 0;**

**}**

**Output:**

**Enter real and imaginary part of 1st complex number:13**

**56.98**

**Enter real and imaginary part of 2nd complex number:34**

**26.48**

**Addition:47+83.46i**

**Subtraction: -21+30.5i**

**Multiplication: -1066.83+2281.56i**

**Division: 1.05042+0.85779i**

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

**6.**An array stores details of 25 students (rollno, name, marks in three subject). Write a program to create such an array and print out a list of students who have failed in more than one subject.

**Program:**

**#include<iostream>**

**using namespace std;**

**struct studentresult{**

**int stuID;**

**string Name;**

**float Phy\_Marks, Chem\_Marks, Maths\_Marks;**

**} studentarray[25];**

**class report{**

**struct studentresult studentarray[25];**

**public:**

**void acceptmarks(){**

**for(int i=0,j=1;i<5,j<=5;i++,j++){**

**cout<<"Enter stuID "<<j<<": ";**

**cin>>studentarray[i].stuID;**

**cout<<"Enter Name "<<j<<": ";**

**cin>>studentarray[i].Name;**

**cout<<"Enter Phy marks: ";**

**cin>>studentarray[i].Phy\_Marks;**

**cout<<"Enter Chemistry marks: ";**

**cin>>studentarray[i].Chem\_Marks;**

**cout<<"Enter Maths marks: ";**

**cin>>studentarray[i].Maths\_Marks;**

**if(studentarray[i].Phy\_Marks<100 && studentarray[i].Chem\_Marks<100 && studentarray[i].Maths\_Marks<100){**

**continue;**

**}**

**else if(studentarray[i].Phy\_Marks>100 || studentarray[i].Chem\_Marks>100 || studentarray[i].Maths\_Marks>100){**

**cout<<"The entered mark is greater than 100"<<endl;**

**acceptmarks();**

**}**

**}**

**}**

**void students(){**

**int passedistinct=0,failed=0,passed=0;**

**for(int i=0;i<5;i++){**

**if(studentarray[i].Phy\_Marks<35 && studentarray[i].Chem\_Marks<35){**

**failed++;**

**cout<<studentarray[i].Name<<endl;**

**}**

**else if( studentarray[i].Chem\_Marks<35 && studentarray[i].Maths\_Marks<35){**

**failed++;**

**cout<<studentarray[i].Name<<endl;**

**}**

**else if(studentarray[i].Phy\_Marks<35 && studentarray[i].Maths\_Marks<35){**

**failed++;**

**cout<<studentarray[i].Name<<endl;**

**}**

**}**

**cout<<"Number of students failed in more than one subject: "<<failed<<endl;**

**}**

**};**

**int main(){**

**report r;**

**r.acceptmarks();**

**cout<<"Printing the name list of students who failed in more than one subject: "<<endl;**

**r.students();**

**return 0;**

**}**

**Output:**

**Enter stuID 1: 1**

**Enter Name 1: hari**

**Enter Phy marks: 23**

**Enter Chemistry marks: 34**

**Enter Maths marks: 56**

**Enter stuID 2: 2**

**Enter Name 2: sunil**

**Enter Phy marks: 34**

**Enter Chemistry marks: 45**

**Enter Maths marks: 67**

**Enter stuID 3: 3**

**Enter Name 3: sundar**

**Enter Phy marks: 34**

**Enter Chemistry marks: 23**

**Enter Maths marks: 56**

**Enter stuID 4: 4**

**Enter Name 4: harry**

**Enter Phy marks: 23**

**Enter Chemistry marks: 34**

**Enter Maths marks: 11**

**Enter stuID 5: 5**

**Enter Name 5: Deming**

**Enter Phy marks: 45**

**Enter Chemistry marks: 56**

**Enter Maths marks: 78**

**Printing the name list of students who failed in more than one subject:**

**Hari**

**Sundar**

**Harry**

**Number of students failed in more than one subject: 3**

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

7. What should be output of below program? program is compiled on g++ compiler.

#include<iostream>

using namespace std;

struct student{

char a; char b; int c;

};

int main()

{

cout<<sizeof(student);

return 0;

}

Options:

(A) 4  
(B) 6  
(C) 8  
(D) 12

Give the Answer**: (C) 8**

**Explanation: Char size is 1 byte and int is of size 4 bytes.**

8. Which of the following statements assigns a value to the hourlyWage member of employee[2}?

Options:

(A) employee[2]->hourlyWage = 50.00;  
(B) employee2.hourlyWage = 7.50;  
(C) hourlyWage[2].employee = 29.75;  
(D) employee[2].hourlyWage = 75.00;

Give the answer: **(D) employee[2].hourlyWage = 75.00;**

9. Which of the following statements outputs the value of the gpa member of element 1 of the student array?

Options:

(A) cout<<student1.gpa;  
(B) cout<<firstStudent.gpa;  
(C) cout<<student[1].gpa;  
(D) cout<<student1 ->gpa;

Give the answer: **(C) cout<<student[1].gpa;**

10. Which of the following statements outputs the value of the gpa member of element 1 of the student array?

Options:

(A) cout<<student1.gpa;  
(B) cout<<firstStudent.gpa;  
(C) cout<<student[1].gpa;  
(D) cout<<student1 ->gpa;

Give the answer: **(C) cout<<student[1].gpa;**