

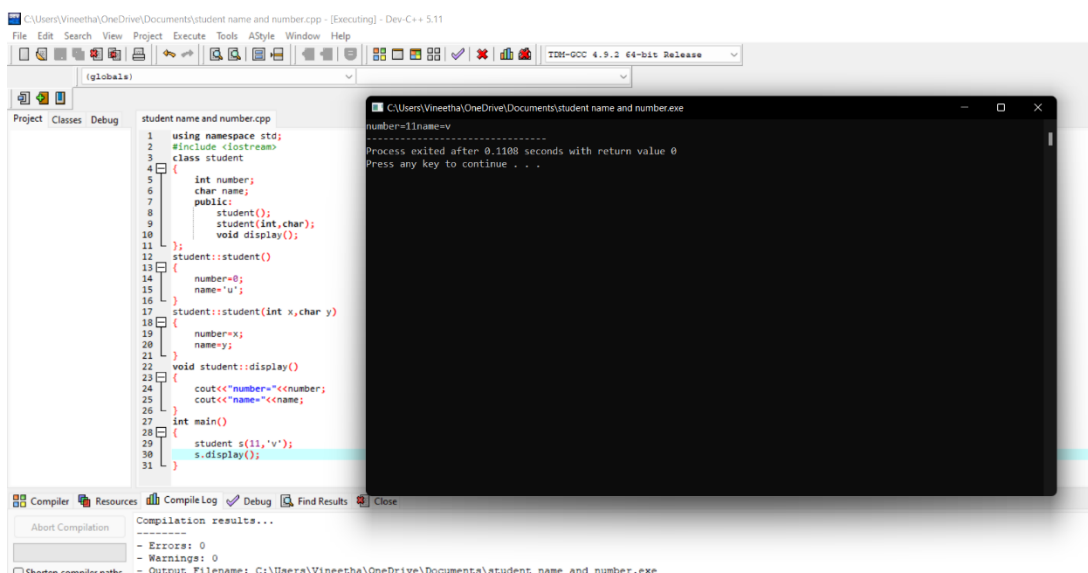
# DSA0136-OBJECT ORIENTED PROGRAMMING WITH C++

DATE:28-09-2022

NAME:B.VINEETHA

REG.NO:192110487

1.WRITE A C++ PROGRAM TO PRINT THE NAME OF THE STUDENTS BY CREATING A STUDENT CLASS.IF NUMBER AND NAME IS PASSED WHILE CREATING THE OBJECTS OF A STUDENTS CLASS;THEN THE NAME SHOULD BE UNKNOWN OTHERWISE THE NAME SHOULD BE EQUAL TO SCREEN VALUE WHILE CREATING THE OBJECT OF THE STUDENT CLASS.



The screenshot shows the Dev-C++ IDE with a C++ program named 'student name and number.cpp' open. The code defines a 'student' class with a 'number' attribute and a 'name' attribute. The 'display()' method prints the number and name. In the 'main()' function, a 'student' object 's' is created with '11' as the number and 'v' as the name. The 'display()' method is called on 's', which outputs 'number=11name=v' to the console. The console window shows the output 'number=11name=v' and a message 'Process exited after 0.1108 seconds with return value 0'. The compilation results window at the bottom shows 'Errors: 0' and 'Warnings: 0'.

```
1 using namespace std;
2 #include <iostream>
3 class student
4 {
5     int number;
6     char name;
7     public:
8         student();
9         student(int, char);
10        void display();
11};
12
13 student::student()
14 {
15     number=0;
16     name='u';
17 }
18 student::student(int x, char y)
19 {
20     number=x;
21     name=y;
22 }
23 void student::display()
24 {
25     cout<<"number="<<number;
26     cout<<"name="<<name;
27 }
28 int main()
29 {
30     student s(11,'v');
31     s.display();
32 }
```

number=11name=v  
Process exited after 0.1108 seconds with return value 0  
Press any key to continue . . .

Compilation results...  
Errors: 0  
Warnings: 0  
Output Filename: C:\Users\Vineetha\OneDrive\Documents\student name and number.exe

**2.CREATE A CLASS NAME RECTANGLE WITH TWO DATAMEMBERS LENGTH AND BREADTH AND A FUNCTION TO CALCULATE THE AREA OF THE RECTANGLE.THIS CLASS CONTAINS 3 CONSTRUCTOR (1)HAVING NO PARAMETER (2)HAVING 2 PARAMETERS (3)HAVING ONLY ONE PARAMETER**

The screenshot shows the Dev-C++ IDE with a project named 'rectangle area using class and object.cpp'. The code defines a `rectangle` class with two data members, `l` (length) and `b` (breadth), and a `display` function. It includes three constructors: a default constructor, a constructor taking two parameters (`l` and `b`), and a constructor taking one parameter (`a`) which sets both `l` and `b` to `a`. The `display` function calculates the area as `l*b` and prints it. The `main` function creates a `rectangle` object `r` with parameters `3` and `6`, and calls `display`.

```

1 class rectangle
2 {
3     int l,b,a,d;
4 public:
5     rectangle();
6     rectangle(int l,int b);
7     rectangle(int a);
8     void display();
9 };
10
11 rectangle::rectangle()
12 {
13     l=0;
14     b=0;
15     a=0;
16 }
17
18 rectangle::rectangle(int x,int y)
19 {
20     l=x;
21     b=y;
22 }
23
24 rectangle::rectangle(int a)
25 {
26     a=a;
27 }
28
29 void rectangle::display()
30 {
31     d=l*b;
32     cout<<"area="<<d;
33 }
34
35 int main()
36 {
37     rectangle r(3,6);
38 }

```

The execution output shows the area calculated as 18.

```

area:18
-----
Process exited after 0.0857 seconds with return value 0
Press any key to continue . . .

```

**3.CREATE A CLASS HAD AMOUNT WITH THE DATA MEMBER IS AMOUNT WITH INITIAL VALUE 50.NOW MAKE TWO CONSTRUCTOR WITH THEIR CLASS I.E.**

**(1)NO PARAMETER I.E. NO AMOUNT IS ADDED**

**(2)HAVING PARAMETER I.E. WHICH IS AMOUNT WILL BE ADDED INTO YOUR ACCOUNT**

**CREATE AN OBJECT FOR HAD AMOUNT CLASS AND DISPLAY FINAL AMOUNT**

The screenshot shows the Dev-C++ IDE with a project named 'addamount.cpp'. The code defines an `addamount` class with two data members, `amount` and `totalamount`, and a `display` function. It includes two constructors: a default constructor that initializes `amount` to 50, and a constructor taking a parameter `x` that sets `amount` to `x`. The `display` function prints the current `amount` and updates `totalamount` by adding the current `amount`. The `main` function creates an `addamount` object `s` and calls `display`.

```

1 using namespace std;
2 #include <iostream>
3 class addamount
4 {
5     float amount,totalamount;
6 public:
7     addamount();
8     addamount(float x);
9     void display();
10 };
11
12 addamount::addamount()
13 {
14     amount=50;
15 }
16
17 addamount::addamount(float x)
18 {
19     amount=x;
20 }
21
22 void addamount::display()
23 {
24     cout<<"amount="<<amount;
25     totalamount+=amount;
26     cout<<"totalamount="<<totalamount;
27 }
28
29 int main()
30 {
31     addamount s(11);
32     s.display();
33 }

```

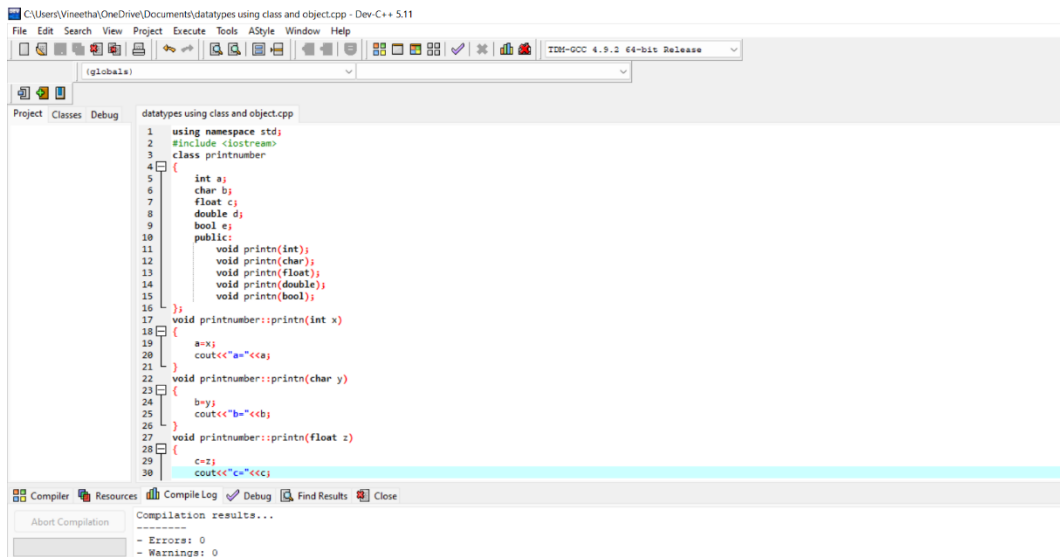
The execution output shows the amount 11 and the total amount 61.

```

amount=11totalamount=61
-----
Process exited after 0.07493 seconds with return value 0
Press any key to continue . . .

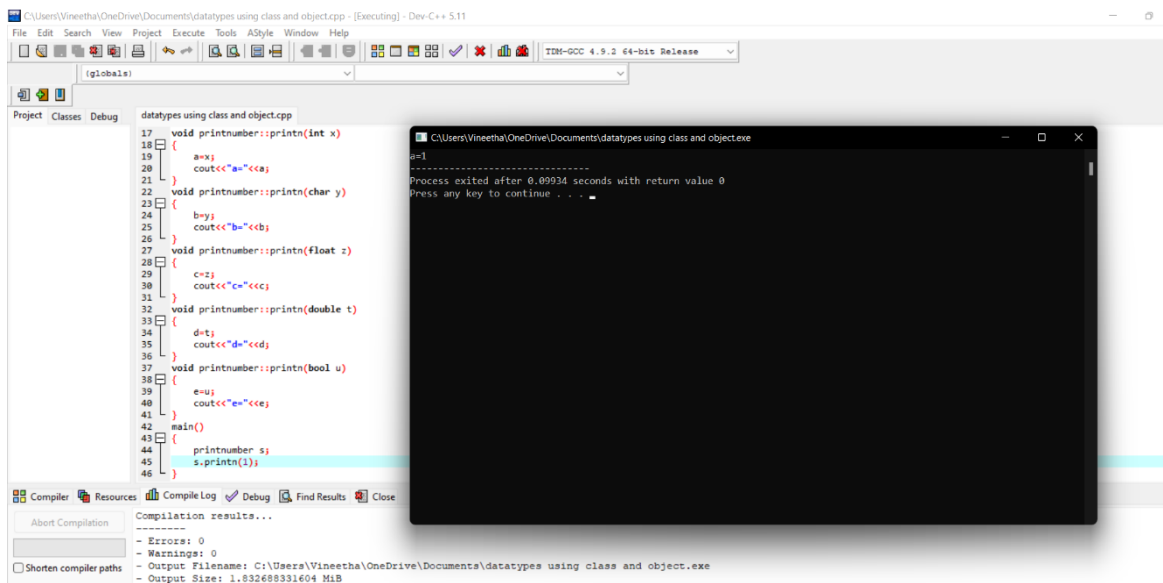
```

#### 4.CREATE A CLASS NAME PRINT NUMBER TO PRINT VARIOUS NUMBERS OF DIFFERENT DATATYPES BY CREATING DIFFERENT FUNCTIONS TO THE SAME NAME PRINT N HAVING A PARAMETER FOR EACH DATATYPE



The screenshot shows the Dev-C++ IDE with the file "datatypes using class and object.cpp" open. The code defines a class named "printnumber" with private variables of various data types (int, char, float, double, bool) and public member functions to print each of them. The compiler output at the bottom shows 0 errors and 0 warnings.

```
1 using namespace std;
2 #include <iostream>
3 class printnumber
4 {
5     int a;
6     char b;
7     float c;
8     double d;
9     bool e;
10 public:
11     void printn(int);
12     void printn(char);
13     void printn(float);
14     void printn(double);
15     void printn(bool);
16 };
17 void printnumber::printn(int x)
18 {
19     a=x;
20     cout<<"a"<<a;
21 }
22 void printnumber::printn(char y)
23 {
24     b=y;
25     cout<<"b"<<b;
26 }
27 void printnumber::printn(float z)
28 {
29     c=z;
30     cout<<"c"<<c;
```



The screenshot shows the Dev-C++ IDE with the same file open, but now the program is being executed. A black console window is overlaid on the IDE, showing the output of the program. The output shows the values of the variables a, b, c, d, and e, each printed on a new line. The console window also shows the process exit message and a prompt to press any key to continue.

```
17 void printnumber::printn(int x)
18 {
19     a=x;
20     cout<<"a"<<a;
21 }
22 void printnumber::printn(char y)
23 {
24     b=y;
25     cout<<"b"<<b;
26 }
27 void printnumber::printn(float z)
28 {
29     c=z;
30     cout<<"c"<<c;
31 }
32 void printnumber::printn(double t)
33 {
34     d=t;
35     cout<<"d"<<d;
36 }
37 void printnumber::printn(bool u)
38 {
39     e=u;
40     cout<<"e"<<e;
41 }
42 main()
43 {
44     printnumber s;
45     s.printn(1);
46 }
```

Compilation results...  
- Errors: 0  
- Warnings: 0  
- Output Filename: C:\Users\Vineetha\OneDrive\Documents\datatypes using class and object.exe  
- Output Size: 1.832688331604 MiB

## 5.CREATE A CLASS TO PRINT AN INTEGER AND CHARACTER USING TRUTH FUNCTIONS HAVING THE SAME NAME BUT DIFFERENT SEQUENCES OF INTEGER AND CHARACTER PARAMETER

```

1  using namespace std;
2  #include <iostream>
3  class printv
4  {
5      int a;
6      char b;
7      public:
8          void print(int,char);
9          void print(char,int);
10         void display();
11     };
12     void printv::print(int x,char y)
13     {
14         a=x;
15         b=y;
16         cout<<"a="<<a;
17         cout<<"b="<<b;
18     }
19     void printv::print(char y,int x)
20     {
21         b=y;
22         a=x;
23         cout<<"a="<<a;
24         cout<<"b="<<b;
25     }
26     int main()
27     {
28         printv k;
29         k.print('v',11);
30     }
    
```

Output:

```

a=11b=v
-----
Process exited after 0.1878 seconds with return value 0
Press any key to continue . . .
    
```

## 6.CREATE A CLASS STUDENT WITH THREE DATAMEMBERS WHICH ARE NAME A AND ADDRESS.THE CONSTRUCTOR OF THE CLASS ASSIGNED DEFAULT VALUES NAMED AS UNKNOWN,NUMBER AS 0, ADDRESS AS NOT AVAILABLE.IT HAS TWO FUNCTIONS WITH SAME AND INFO.THE FIRST FUNCTION IS TWO PARAMETERS:NAME AND AGE AND SECOND IS 3 PARAMETERS:NAME,AGE AND ADDRESS RESPECTIVELY.PRINT THIS DETAILS OF 10 STUDENTS

```

1  using namespace std;
2  #include <iostream>
3  class student
4  {
5      char name[20],address[20];
6      int age;
7      public:
8          void getdata();
9          void display();
10     };
11     void student::getdata()
12     {
13         cout<<"\nenter the student name-";
14         cin>>name;
15         cout<<"\nenter the student age-";
16         cin>>age;
17         cout<<"\nenter the student address-";
18         cin>>address;
19     }
20     void student::display()
21     {
22         cout<<"\nthe entered student details below= ";
23         cout<<"\nname:\n"<<name;
24         cout<<"\nage:\n"<<age;
25         cout<<"\naddress:\n"<<address;
26     }
27     int main()
28     {
29         student s[10];
30         for(int i=0;i<10;i++)
    
```

Output:

```

-----
- Errors: 0
- Warnings: 0
    
```

