

Final - Term Paper

Object oriented programming

Name: ASHFAQ AHMAD

Reg No: 19PWCSE1795

Section : B

Date : 11-03-2021

— xx — xx — xx — xx

S M T W T F S

Date:.....

Q No (1)C++ Program

```
#include <iostream>
using namespace std;
```

```
class bank account
```

```
{ Private:
```

```
    char name[20];
```

```
    char account-type[20];
```

```
    int account-number;
```

```
    int balance;
```

```
Public:
```

```
void initialize()
```

```
{
```

```
    cout << "\n enter Account Holder  
    name:";
```

```
    gets(name);
```

```
    cout << "\n enter Account type:";
```

```
    gets(account-type);
```

```
    cout << "\n enter account no:";
```

```
    cin >> account-number;
```

```
    cout << "\n enter balance to  
    deposit:";
```

```
    cin >> balance;
```

```
}
```

```
void deposit()
```

```
{
```

```
    int bal;
```

```
    cout << "\n enter the amount  
    to deposit:";
```

```
    P    +    T    +    0
```

Castelli

```

cin >> bal;

balance += bal;
cout << "\n Amount deposited
        Successfully \n Your New
        balance:" << balance;
}

void check()
{
    int bal;
    cout << "\n Your balance:"
    << balance << "\n enter amount
    to withdraw:";

    cin >> bal;
    if (bal <= balance)
    {
        balance = bal;
        cout << "\n Remaining balance:"
        << balance;
    }
    else
    {
        exit(0);
    }
}

void display()
{
    cout << "\n Name:";
    puts(Name);
    cout << "\n balance:" << balance;
}

}
} → class END

P T P O

```

S M T W T F S

Date:.....

```

void main ()
{
    int i;
    classcr ();
    Bank bk;
    bk.initialize();
    cout << "\n 1. Your Information\n2.
    Deposit\n3. withdraw\n enter
    your choice\n";
    cin >> i;
    if (i == 1)
    {
        bk.display();
    }
    else if (i == 2) {
        bk.deposit(); }
    else if (i == 3) {
        bk.check(); }

    getch();
}

```

xx ——— x x ——— x x ———

P T P O

Q No 2

```
#include <iostream>
using namespace std;
class CFloat
{
    Private:
        float a;
    Public:
        CFloat (float c)
        {
            a = c;
        }
        float operator + (float b)
        {
            return a + b;
        }
        float operator - (float b)
        {
            return a - b;
        }
        float operator * (float b)
        {
            return a * b;
        }
        float operator / (float b)
        {
            return a / b;
        }
};
```

→ (Class End)

int main()

```
{
    CFloat c(22.41);
    float d = c + 40.5;
}
```

P T F O

Castelli

```

cout << d << endl;
d = c - 40.5;
cout << d << endl;
d = c * 40.5;
cout << d << endl;
d = c / 40.5;
cout << d << endl;

return 0;
}

```

Q No 3:

```

#include <iostream>
#include <conio.h>
#include <math.h>

```

```

class Rec;
class Polar
{
    float a, r;
public:
    polar ()
    {
        a = 0;
        r = 0;
    }
    polar (float a1, float r1)
    {
        a = a1;
        r = r1;
    }
    void putData()

```

```

}
    cout << "\n Polar Data";
    cout << "\n A = " << a << ", R = "
        << r << " ";
}
float geta()
{
    return a;
}
float getr()
{
    return r;
}
} (class END)

```

Class Rec

```

{
    float x, y;
public:
    Rec()
    {
    }
    Rec(float x1, float y1) {
        x = x1;
        y = y1;
    }
    void PutData()
    {
        cout << "\n Rect Data";
        cout << "\n x = " << x << ", y = "
            << y << " ";
    }
    Rec(Polar P) {
        x = P.getr() * cos(3.14 * P.geta() / 180);
        y = P.getr() * sin(3.14 * P.geta() / 180);
    }
    P + T + 0
}

```


S M T W T F S

Date:

```
friend Rect operator +(Rect &r1,
                        Rect &r2)
```

```
{
```

```
    Rect r3;
```

```
    r3.x = r1.x + r2.x;
```

```
    r3.y = r1.y + r2.y;
```

```
    return r3;
```

```
}
```

```
operator polar()
```

```
{
```

```
    return polar(atan(x/y), sqrt(x*x
                                + y*y));
```

```
}
```

```
};
```

```
int main()
```

```
{
```

```
    clrscr();
```

```
    float p1, p2;
```

```
    polar p2(4, 5);
```

```
    polar p1(2, 3);
```

```
    polar p3;
```

```
    Rect r1, r2, r3;
```

```
    r1 = p1;
```

```
    r2 = p2;
```

```
    r3 = r1 + r2;
```

```
    p3 = r3;
```

```
    r3.putData();
```

```
    p3.putData();
```

```
    getch();
```

```
    return 0;
```

```
}
```

P T P O

Castelli

Output:

Rec Data

$x = 7.98601$, $y = 0.453251$

Polen Data

$A = 1.5141$, $R = 7.99886$.

— xx — xx — xx — xx

CP NO \Rightarrow 4

Include <iostream>

using namespace std;

class Shape

{

public: double a, b;

void get_data()

{

cin >> a >> b;

}

virtual void display_area()=0;

};

class Triangle: public Shape

{

public: void display_area()

{

cout << "Area of Triangle"

<< $0.5 * a * b$ << endl;

}

};

P = Tpo

```

class Rectangle: public Shape
{
    public: void display_area()
    {
        cout << "Area of Rectangle"
              << a * b << endl;
    }
};

```

```

int main()
{

```

```

    Triangle t;
    Shape *st = &t;

```

```

    Shape *st = &t;
    cout << "enter base & altitude:";
    st->get_data();
    st->display_area();

```

```

    Rectangle r;
    Shape *sr = &r;

```

```

    cout << "enter length and breadth:";
    sr->get_data();
    sr->display_area();

```

```

    return 0;
}

```

Output:

```

Enter base & altitude: 20 10
Area of Triangle 100
Enter length and breadth: 25 15
Area of rectangle 375

```

xx ——— xx ——— xx ———

S M T W T F S

Date:

Q No = 5

Q5 part 1:
long double x;

Q5 Part 2:
char *cp = ((char *)vp);

Q5 Part 3:
Correct

Q5 Part 4:
int *sp = new int[1];

Q5 Part 5:
enum { green, yellow, red};

Q5 Part 6:
Correct

Q5 Part 7:
const int array_size = 840;

Q5 Part 8:
for (int i = 1; i < 10; i++) {
cout << i << "\n"; int number = 100;
float *p = new float [101]; }

Q5 part 9:
int public i = 1000;

Q5 Part 10:
Correct

xx — xx — xx — xx —

P r T + o Castelli

Q No \Rightarrow 6:

```
#include <iostream>
#include <string.h>
using namespace std;
```

```
class cricket {
private:
    char * name, * teamname;
    double score, age;
public:
    cricket ()
    {
        score = 0;
        age = 0;
    }
```

```
    cricket (char * n, char * t, double s,
             double a)
    {
        int namelen = strlen(n);
        name = new
        char [namelen + 1];
        int tnamelen = strlen(t);
        teamname = new
        char [tnamelen + 1];
        strcpy (name, n);
        strcpy (teamname, t);
        score = s;
        age = a;
    }
```

```
    cricket (cricket & c)
```

```
    {
        score = c.score;
        age = c.age;
    }
```

S M T W T F S

Date:

```

int namelen = strlen(c.name);
name = new
char[namelen+1];
int tnamelen = strlen(c.teamname);
teamname = new char[tnamelen+1];
strcpy(name, c.name);
strcpy(teamname, c.teamname);
}
void display()
{
    cout << "Name: " << name << "
    teamname" << teamname << "
    Score" << score << "age = "
    << age << endl;
}
~ Cricket() {
    delete name;
    delete teamname;
}
}
}

```

```

int main()
{
    char p1[7] = "M Shahid SHAHID
    Afridi";

```

```

    char p2[7] = "Hafeez";

```

```

    char t1[7] = "Pakistan";

```

```

    char t2[7] = "India";

```

```

    Cricket c1(p1, t1, 4, 46),
    c2(p2, t2, 22, 50);

```

Castelli

C1.display();

C2.display();

C1 = C2;

C1.display();

return 0;

}

Output:

Name: SHAHID AFRIDI

Teamname: Pakistan

Score : 4

age = 46

Name: Hafeez

teamname: India

Score : 22

age: 50

Assignment operator output:

Name: Hafeez

teamname: India

Score : 22

age: 50

x x x x x x x x

The END

ASHFAQ AHMAD

Reg No:
19pwe56
1795

Castelli