

Polymorphism is used in following programme

output

@	@	@	@
9	18	27	

Q4. No, it will not run without any error
 ∴ function definition is not given for
 one of the prototypes declared in class Test.
 Moreover, no value is returned in main();
 correct programme:

```
#include <iostream.h>

class Test
{
public:
    void f(int n, double m);
    void f(int n, int m);
};

void Test::f(int n, double m) {
    cout << "Addition of int and double value resulted in : "
         << (double)(n+m) << endl;
}

void Test::f(int n, int m) {
    cout << "Addition of two int values resulted in : "
         << n+m << endl;
}

int main()
{
    Test t1;
```

Polymorphism is used in following programme

output

@	@	@	@
9	18	27	

Q4. No, it will not run without any error
 ∴ function definition is not given for
 one of the prototypes declared in class Test.
 Moreover, no value is returned in main();
 correct programme:

```
#include <iostream.h>

class Test
{
public:
    void f(int n, double m);
    void f(int n, int m);
};

void Test::f(int n, double m) {
    cout << "Addition of int and double value resulted in : "
         << (double)(n+m) << endl;
}

void Test::f(int n, int m) {
    cout << "Addition of two int values resulted in : "
         << n+m << endl;
}

int main()
{
    Test t1;
```

Polymorphism is used in following programme

output

@	@	@	@
9	18	27	

04. No, it will not run without any error
 ∴ function definition is not given for
 one of the prototypes declared in class Test.
 Moreover, no value is returned in main();
 correct programme

```
#include <iostream.h>
```

```
class Test
```

```
{
public:
    void f(int n, double m);
    void f(int n, int m);
}
```

```
void Test::f(int n, double m) {
```

```
cout << "Addition of int and double value resulted in :"  

    << (double)(n+m) << endl;
```

```
void Test::f(int n, int m)
```

```
{
    cout << "Addition of two int values resulted in :"  

    << n+m << endl;
}
```

```
int main()
```

```
{
    Test t1;
```